

Una caminada pels principis de disseny

Josep M. Merenciano

Departament de Ciències de la Computació
Universitat Politècnica de Catalunya

`meren@cs.upc.edu`

Febrer de 2017

3 de febrer de 2017

Taula de continguts

<i>Gastem la sola: una bonica manera de conèixer el país</i>	vii
Llegenda	ix
1 Elements bàsics	ix
2 Referències creuades	x
3 Canvis en les versions	xi
 I Preparem el viatge	 1
1 Principis aplicables al disseny	5
1 Principis de modelització	8
2 Principis de desenvolupament de software	10
3 Regles i principis	16
Principis	18
Definicions	18
2 Disseny: punt de partida i objectius	19
1 El punt de partida del disseny	22
2 Objectiu bàsic del disseny	23
3 Les tasques del dissenyador	24
4 Assignació de responsabilitats	25
Principis	29
3 Elements del disseny	31
1 Elements estructurals	36
2 Àmbit de validesa dels enllaços dirigits	44
3 Visibilitat	52
4 Col·laboració	73
5 Modelització de les associacions del MC	75
Definicions	78
4 Models en el disseny	79
1 El model col·laboratiu	81
2 Model de components (MComp)	81
3 El perill de l'enginyeria inversa	83
4 Consistència dels models	84
5 Construcció del MComp	85

5	Multiobjectes	87
1	Presentació dels <i>multiobjectes</i>	90
2	Col·laboració i multiobjectes	92
3	El multiobjecte com a artifici	96
4	Incursió en la implementació	97
5	MOMO: Un modisme pels multiobjectes	100
	Principis	102
	Definicions	102
II	Comencem a caminar	103
6	Primers principis de disseny	107
1	L'expertesa és un grau	110
2	El paper dels controladors	112
3	Necessitat de comunicació entre els ES	113
4	Responsabilitats dels controladors	115
	Principis	116
7	<i>GLS</i> Cas d'ús <i>ferInscripció</i>	117
1	Comencem a dissenyar	119
2	Contracte de <i>novaInscripcio (c:Caminada)</i>	119
3	POST: Crear una nova <i>i:Inscripció</i>	120
4	Anàlisi d'alternatives	123
8	Anàlisi d'alternatives	125
1	Necessitat dels principis avaluatius	127
2	Cohesió	127
3	Acoblament	128
4	Anàlisi de l' <i>Encarrilament</i>	128
	Principis	130
	Definicions	130
9	Una patata que crema	131
1	<i>GLS</i> Tasca pendent	135
2	El problema de la patata calenta	135
3	Ús, emmagatzematge i creació	140
4	Repositoris	147
	Definicions	150
10	Principis de creació	151
1	Principi <i>Creador (GRASP)</i>	158
2	Principi <i>Expert delegat</i>	159
3	Principi <i>Creador orb</i>	161
4	Emmagatzematge	164
5	Principi del <i>Creador canònic</i>	173
6	Emmagatzematge primari	176
7	Principi del <i>Creador generalitzat</i>	182
8	Migració i compartició de realitzacions	183
9	Ús dels diferents principis de creació	187

TAULA DE CONTINGUTS

J.M. Merenciano

10 La patata calenta i els principis de creació	189
11 Creació i recuperació de la informació	191
Principis	197
Definicions	197
11 ^{GLS} L'ES novaInscripció	199
1 La i:Inscripció com a patata calenta	202
2 POST: Enllaçar i i c	215
12 Arguments implícits	217
1 Realització activa	220
2 Manteniment consistent dels contractes	224
3 Parèntesi sobre l'eficiència	227
Principis	229
Definicions	229
13 ^{GLS} POST: La i:Inscripció es manté activa	231
1 Activació de la i:Inscripció	235
2 Activació a M1	235
3 Activació a M2	239
4 Anàlisi de l'eficiència	242
5 Conclusions i decisions de disseny	245
6 Els models en la situació actual	247
III Terres pantanoses	249
14 Encontres en la tercera fase	253
1 Esbós del camí	257
2 Un problema comunicatiu	257
3 Mons comunicables	259
4 Un llenguatge comú	263
5 Comunicació i principis de modelització	267
6 La comunicació és possible	275
Definicions	276
15 Identificadors	277
1 Esbós del camí	282
2 Llenguatge per nivells	282
3 Propietat d'identificació	288
4 Component ben definit	296
5 Identificació i llenguatge de comunicació	298
6 La comunicació és possible	313
Definicions	315
IV Construïm ponts	317
16 Identificació robusta	321
1 Recuperació de realitzacions	323
2 Identificació interna i externa	326

3	Els identificadors en els ES	327
4	Contracte robust	329
	Principis	331
	Definicions	331
17	Flux comunicatiu	333
1	Arquitectura de tres capes	337
2	Flux comunicatiu extern	340
3	El flux comunicatiu intern	348
4	Encaixada	352
	Principis	356
	Definicions	356
18	^{GLS} Identificadors i novaInscripció (caminada)	357
1	La caminada com a identificador	357
2	La caminada com a objecte	359
19	Àmbits d'identificació	363
1	Identificació i espai d'identificació	366
2	Els identificadors en el MC	367
3	Els identificadors en el MComp	374
4	Identificadors i repositoris	375
5	Identificació atributiva	379
6	Visibilitat dels àmbits d'identificació	381
	Principis	383
	Definicions	383
20	Generació d'identificadors	385
1	Mecanismes de generació	391
2	Introducció a la GPC	392
3	GPC amb un sol repositori	399
4	GPC amb múltiples repositoris	409
5	Generació per comprovació: conclusions	413
6	Generació única	413
7	Principi de la <i>Individualització efectiva</i>	416
	Principis	418
	Definicions	418
21	Identificació: conclusions	419
1	Identificadors: conclusions	421
2	^{GLS} La identificació d'Inscripcions	424
22	^{GLS} Contracte de l'ES nouParticipant (nom, edat)	425
1	Contracte de partida	428
2	Àmbits d'identificació	429
3	Desenvolupament àgil	436
4	Ús de la <i>Presentació</i>	437
5	Conclusions sobre el contracte	439
6	Retorn a l'especificació	440

V	Seguim explorant	443
23	Modelitzacions no bàsiques	447
1	Modelització d'un concepte associatiu	451
2	Conceptes febles	466
3	Associacions ternàries	469
24	^{GLS}La idea de <i>participació</i>	471
1	El concepte <i>Persona</i>	475
2	El concepte <i>Participant</i>	476
3	El concepte <i>Participació</i>	478
4	El concepte <i>Línia d'inscripció</i>	483
5	Conclusions sobre la <i>participació</i>	486
25	^{GLS}CU <i>ferInscripció</i> (2)	489
1	Punt de partida	492
2	Controlador	493
3	POST: Creació de <i>li:Línia</i>	493
4	POST: Enllaç entre <i>i i li; i activa</i>	501
5	Esdeveniment <i>fiInscripcio()</i>	501
26	Compleció semàntica	503
1	^{GLS} Un exemple: la semàntica de <i>fiInscripcio()</i>	506
2	El problema de la compleció semàntica	506
3	Principi de la <i>Incorporació tardana</i>	514
27	^{GLS}El CU <i>ferInscripció</i> amb <i>Incorporació tardana</i>	519
1	Els contractes fins al moment	522
2	Anàlisi de la <i>Incorporació tardana</i>	523
3	Un disseny amb <i>Incorporació tardana</i>	524
28	^{GLS}Cas d'ús <i>Creació de dorsals</i>	531
1	Esdeveniment <i>activaCaminada(nomC)</i>	533
2	Esdeveniment <i>creaDorsalsGrup(id)</i>	535
	Historial de versions	541
	Principis	543
	Definicions	544

Gastem la sola: una bonica manera de conèixer el país

Gastem la sola és una iniciativa particular sense ànim de lucre que té per objecte facilitar el coneixement del país a tots els qui estiguin disposats a caminar. La idea es basa en catalogar, adaptar i senyalitzar camins que han de ser fets a peu; i en organitzar caminades populars multitudinàries. Per poder tirar endavant la iniciativa es necessita un sistema de gestió d'aquestes caminades.

Qualsevol persona, nen o animal de companyia pot apuntar-se a una caminada (o a tantes com vulgui) dins dels terminis d'inscripció d'aquesta. Per inscriure's n'hi ha prou en donar el nom complet i l'edat. La inscripció es pot fer en qualsevol dels centres associats que hi ha arreu del país, i pot ser individual o en grup. Com a resultat de la inscripció hom rep un resguard amb la data de la inscripció, el nom de la caminada on s'ha fet la inscripció, i el llistat de tots els components del grup (nom i edat), amb indicació de quin d'ells n'és el responsable. El responsable ha de ser un major d'edat, i ens ha de donar un telèfon mòbil de contacte.

En el moment de començar una caminada, els inscrits es presenten al punt de control pertinent, on els lliuren un dorsal amb un número i una seqüència de lletres; el número és únic, la seqüència de lletres, que coincideix amb l'identificador del resguard de la inscripció, identifica el grup de participants. A més, el color del dorsal indica si majoritàriament el grup és de nens, d'adults o de jubilats. El responsable de cada grup porta el dorsal tramat.

Cada caminada té un nom, una data de realització, i un període d'inscripció. No hi ha cap límit en el nombre de participants.

Les caminades es componen d'etapes. Les etapes s'identifiquen amb un codi; cada etapa té associada un indret singular d'origen i un altre de destinació. Les etapes es defineixen de manera independent a les caminades; una caminada pot repetir etapes, i una etapa pot aparèixer en més d'una caminada.

Gastem la sola té un catàleg de camins. Cada camí queda descrit pels indrets singulars on comença i on acaba el camí. Per exemple, el Camí de la Font Vella és el camí que comença sota el Pi de l'Òliba, i acaba a la Font Vella, tot passant pel Pas de Barca. Ara bé, diferents camins poden tenir el mateix origen i destinació. Per exemple, el Camí Llarg és el que comença sota el Pi de l'Òliba i acaba a la Font Vella, però enlloc de travessar el riu pel Pas de Barca va a donar la volta pel Pont de les Tres Aigües.

L'estructuració en etapes ve a ser com un disseny a grans trets del que ha de ser la caminada: s'indiquen els indrets singulars mínims pels que cal passar (i on hi haurà controls), i en quin ordre. El que no diu és quins camins s'han de seguir per passar per tots aquests indrets. La realització canònica és la proposta de camins que *Gastem la sola* fa per cada caminada. En ells se'ns indica la seqüència de camins que cal seguir, quina és la durada aproximada amb què s'ha de recorre cada camí, i la duració aproximada de l'etapa dins la caminada.

Malgrat la proposta que fa *Gastem la sola* en forma de realització canònica, cada participant pot seguir els camins que li semblin més convenients. Per tal de millorar les propostes de realitzacions canòniques, *Gastem la sola* demana als participants que de manera optativa indiquin quins camins han seguit realment, i quan de temps els ha portat cada etapa.

En una realització (canònica o no) cada etapa es pot compondre de diversos camins; un mateix camí pot usar-se en diferents etapes, de la mateixa caminada o d'una altra.

Gastem la sola disposa d'una sèrie de patrocinadors que ofereixen diferents records per repartir entre els participants. Per exemple, Turisme Feudal ha donat a *Gastem la sola* 100 postals del castell de Montfort i 50 clauers amb la creu templària; i al seu torn, Xino Xano ha regalat 500 adhesius amb el seu anagrama.

En arribar al final de la darrera etapa d'una caminada, cada participant rep com obsequi un dels records que *Gastem la sola* ha obtingut dels seus patrocinadors. En el moment de lliurar el record s'intenta que cap participant rebi el mateix obsequi que un altre membre del grup; i en cas de ser possible s'intenta que tots els participants d'un mateix grup rebin records donats per patrocinadors diferents.

Podem assumir que en el moment de començar una caminada *Gastem la sola* té prou obsequis per tots els participants; el que pot ser és que a mesura que aquests vagin acabant la caminada alguns obsequis es vagin esgotant.

Llegenda

1 Elements bàsics

Definició. Això és una definició

Definició de dels temres tècnics emprats

Aquest és un principi. Enunciació del principi

Principi



Conclusió o resum de l'anàlisi feta.

(1)

Observació. Aspectes a tenir presents.

(1) !

Pregunta. Planteja una pregunta que els següents paràgrafs intentaran resoldre.

(2) ?

Assumpció. Suposició que mantenim com a certa.

(3) ? !

Alternativa. Punt de brancament. D'entre les diferents possibilitats de decisió, ens decidim per l'alternativa exposada. Òbviament aquest és un punt on aplicar la reconsideració.

(4) ➔

H (5)

Hipotesi. Hipòtesi sobre la que es basen el plantejaments.

Exemple 1 (Un exemple) *Cas concret que exemplifica el que s'està explicant.*

Excursió. (Una excursió) A vegades hi ha explicacions, que tot i ser pertinents en el moment que es donen, s'allunyen del fi argumental i en introduir-les seria fàcil perdre'ns per les branques dels raonaments.

Altres vegades el fil argumental permet enllaçar amb alguns aspectes que encara no s'han vist. No és el moment doncs d'encarar-nos-hi, però tampoc es tracta de perdre l'oportunitat.

Tant en un cas com un altre usem les *excursions* per fer les disquisicions pertinents.

Per tant, una *excursió* es pot veure com una discussió que, tot i ser pertinent, no és necessària per a poder seguir els raonaments, i que de ben segur hom es pot saltar en una primera lectura

2 Referències creuades

Referències al marge. Les referències creuades s'indiquen en el text amb el superíndex \rightarrow , i amb una nota al marge.

Referències a un capítol. En les referències creuades a un capítol (1.Principis aplicables al disseny, pàgina 5) \rightarrow) indiquem el número del capítol i el seu títol.

1.Principis aplicables al disseny, pàgina 5

Referències a un apartat. En les referències creuades a un apartat o capítol (1.1.3.Modelització continua, pàgina 10) \rightarrow) el primer dígit indica el capítol. Cal tenir present, però, que en el redactat les capçaleres dels apartats no inclouen pas el capítol: en aquest cas la referència és a l'apartat *Modelització continua* que és l'apartat 1.3 del primer capítol (*Principis aplicables al disseny*)

1.1.3.Modelització continua, pàgina 10

Referències a un element numerat. En les referències creuades a un element numerat (pregunta 2) \rightarrow) només indiquem el valor del comptador. Si l'element numerat porta algun títol (exemple *Un exemple*) \rightarrow) també el fem aprèixer en la referència.

Pregunta 2, pàgina ix

Exemple 1.Un exemple, pàgina x

Numeració dels elements. Els *exemples* i els *conclusions* tenen cadascun d'ells el seu propi comptador. Per contra, la resta d'elements (observacions, preguntes, assumpcions, alternatives, hipòtesis) comparteixen un mateix comptador. La idea subjacent és que els exemples i les conclusions tenen entitat pròpia i tenen sentit en qualsevol moment. En canvi els altres elements poden formar part d'un raonament o d'un fil argumental i poden quedar obsoletes: preguntes ja resoltes; alternatives reconsiderades; etc.

3 Canvis en les versions

Canvi petit. *.Canvi menor* Petites millores en el redactat o petits afegint ens el contingut.

Canvi important. *.Canvi important* Canvi que canvia la manera de veure les coses, o bé l'estructura del document (o algun dels seus capítols)

Novetat. *.Novetat* Afegit d'un tema totalment nou

Part I

Preparem el viatge

Part I: Taula de continguts

1	Principis aplicables al disseny	5
1	Principis de modelització	8
2	Principis de desenvolupament de software	10
3	Regles i principis	16
	Principis	18
	Definicions	18
2	Disseny: punt de partida i objectius	19
1	El punt de partida del disseny	22
2	Objectiu bàsic del disseny	23
3	Les tasques del dissenyador	24
4	Assignació de responsabilitats	25
	Principis	29
3	Elements del disseny	31
1	Elements estructurals	36
2	Àmbit de validesa dels enllaços dirigits	44
3	Visibilitat	52
4	Col·laboració	73
5	Modelització de les associacions del MC	75
	Definicions	78
4	Models en el disseny	79
1	El model col·laboratiu	81
2	Model de components (MComp)	81
3	El perill de l'enginyeria inversa	83
4	Consistència dels models	84
5	Construcció del MComp	85

5 Multiobjectes	87
1 Presentació dels <i>multiobjectes</i>	90
2 Col·laboració i multiobjectes	92
3 El multiobjecte com a artifici	96
4 Incursió en la implementació	97
5 MOMO: Un modisme pels multiobjectes	100
Principis	102
Definicions	102

Capítol 1

Principis aplicables al disseny

1 Principis de modelització	8
1.1 Model i 'realitat'	8
1.2 El nom i la cosa	9
1.3 Modelització continua	10
2 Principis de desenvolupament de software	10
2.1 Encapsulament	11
2.2 Ocultació	11
2.3 Interfície	12
2.4 Contractes	12
2.5 Desenvolupament sota contracte	14
2.6 Caixa blanca i caixa negra	14
2.7 Desenvolupament amb caixes negres	15
2.8 Desenvolupament sota contracte amb caixes negres	15
3 Regles i principis	16
Principis	18
Definicions	18

Contingut detallat del capítol 1

1 Principis de modelització	8
1.1 Model i 'realitat'	8
Focalització, abstracció i simplicitat	8
Multiplicitat de models	8
Espill	8
Model especular	8
1.2 El nom i la cosa	9
Franquícia	9
Denominació consistent amb la semàntica de la franquícia	9
Franquícia obligada	9
Continuïtat de denominació	9
Referent únic	9
Concreció, claredat, no ambigüitat	9
1.3 Modelització continua	10
Modelització continua	10
Modelització especular amb franquícia obligada	10
Continuïtat entre la "realitat" i el model	10
Traçabilitat de la modelització	10
2 Principis de desenvolupament de software	10
2.1 Encapsulament	11
Encapsulament	11
2.2 Ocultació	11
Ocultació	11
Dues perspectives: ús i construcció	11
Dos actors: client i servidor	11
Perspectiva i coneixement	11
Afectació local dels canvis	11
2.3 Interfície	12
Interfície obligada	12
2.4 Contractes	12
Insuficiència de les interfícies	12
Manual del bon ús	12
Contractes PRE/POST	13
Lectura d'un contracte PRE/POST	13
2.5 Desenvolupament sota contracte	14
Desenvolupament sota contracte	14
Divisió de responsabilitats; confiança mútua	14
2.6 Caixa blanca i caixa negra	14
Origen de la nomenclatura	14
Ocultació total	14
Coneixement limitat a la interfície	14
Principis que porten a la caixa negra	15
2.7 Desenvolupament amb caixes negres	15

2.7.1	Independència entre l'ús i la construcció	15
	Dualitat client/servidor fins a l'extrem	15
	Ús amb desconeixement	15
	Construcció aïllada	15
2.8	Desenvolupament sota contracte amb caixes negres	15
	El contracte com a mecanisme de comunicació	15
	Desenvolupament independent	16
	Algunes conseqüències	16
3	Regles i principis	16
	En el disseny regnen els principis	16
	Regles en el disseny	16
	Principis	18
	Definicions	18

Tot seguit presentem resumidament els principis que tot i no ser propis del disseny hi són plenament aplicables.

1 Principis de modelització

1.1 Model i 'realitat'

Definició. Model

Representació de les propietats rellevants d'allò que volem estudiar.

Focalització, abstracció i simplicitat. Tot model es basa en els principis d'abstracció, simplicitat i focalització. El model només intenta expressar allò que ens interessa pel nostre estudi (*focalització*); ho fa en els termes més genèrics possibles, oblidant-se de les circumstàncies o contextos particulars (*abstracció*); i cercant sempre la representació més simple (*simplicitat*).

Multiplicitat de models. Una mateixa "realitat" tindrà tants models com objectius tinguem sobre aquesta realitat: cada model focalitza sobre objectius diferents.¹

Principi



Espill. En la construcció del model ens emmirallem en la "realitat" modelitzada. És a dir, *intensem* que cadascun dels conceptes, idees, objectes, etc. es corresponguin *un a un* amb conceptes, idees, etc. de la "realitat" modelitzada.

Model especular. Sovint parlem de *model especular* per referir-nos a un model que segueix el principi de l'*Espill*.

Exemple 2 (Quadre a l'oli) *Un quadre a l'oli és una representació d'allò que estem pintant. Si pintem una casa amb el seu jardí, la "realitat" casa la modelitzarem amb un element del quadre diferent a l'element que usem per modelitzar la "realitat" arbre. En el quadre tenim diferents formes, de diferents colors, i cadascuna d'elles és la representació d'algun dels objectes que estem pintant. El quadre és una visió especular d'allò que estem pintant.*

Exemple 3 (Pintura puntillista) *Aquest és un exemple de model no especular (no seguim el principi de l'Espill). La "realitat" que volem modelitzar només la podem copsar si considerem el quadre en la seva totalitat. No hi ha cap forma, ni cap color que representi la casa; no hi ha cap forma*

¹Posem "realitat" entre cometes perquè res impedeix fer models de creacions abstractes, com per exemple un altre model. Així distingim entre la "realitat" modelitzada i el model que l'expressa.

1.1 Principis de modelització

J.M. Merenciano

ni cap color que representi els arbres. Només quan contemplem el quadre a certa distància, l'amalgama de punts de color, d'aparença totalment caòtica, s'ordenen i presenten una visió on apareix la casa i els arbres.

1.2 El nom i la cosa

Franquícia. En el moment de construir un model *podem* usar per anomenar els elements del model noms que es corresponguin amb elements de la “realitat” modelitzada, sempre i quan hi hagi un cert *correlat semàntic* entre els elements que comparteixen nom.

Principi



Denominació consistent amb la semàntica de la franquícia. El préstec o lloguer d'un nom significa que aquest nom passa a tenir més d'un significat: anomena un element del model, però també anomena un element de la “realitat” modelitzada. Per evitar mals entesos el préstec només es pot fer sota el *contracte de la franquícia*: si usem un nom de la “realitat” en un element del model és perquè aquest element d'alguna manera modelitza alguns dels aspectes d'allò que en la “realitat” modelitzada es diu de la mateixa manera. En d'altres paraules: la denominació ha de ser semànticament consistent.²

Franquícia obligada. Si en el model *podem* usar un nom de la “realitat” modelitzada (principi de la *Franquícia*), l'hem d'usar.

Principi



Continuïtat de denominació. La franquícia obligada té una conseqüència important: existeix *continuïtat* en els noms dels diferents nivells de modelització o representació, és a dir, tenim un mateix nom per tots els elements que en darrera instància modelitzen la mateixa “realitat”

Referent únic. *L'homonímia és sinonímia.* És a dir, tots aquells noms idèntics (homònims) necessàriament s'han de referir a una mateixa “realitat” (són sinònims), independentment d'on s'usin, si en la “realitat” modelitzada, si en el model, si en el model del model, etc.

Principi



Concreció, claredat, no ambigüitat. El principi del *Referent únic* recolza en els objectius de concreció, claredat i no ambigüitat. El model és un mitjà per expressar i comunicar un coneixement. Si admetem que un mateix nom pot significar “realitats” diferents (subversió del principi del *Referent únic*) llavors ens cal afegir el context desambiguador necessari per fer la comunicació efectiva (amb la conseqüent pèrdua de claredat o concreció).

²Una franquícia de sabates només pot vendre sabates, i no pot pas vendre plàtans. En usar el nom de la franquícia ens comprometem a fer servir aquest nom només per allò que el contracte de la franquícia permet.

1.3 Modelització continua

Principi



Modelització continua. Els noms i elements usats en el model s'han de correspondre a noms i elements de la "realitat" modelitzada.

Modelització especular amb franquícia obligada. La *franquícia obligada* ens diu que *hem* de reusar noms allà on sigui possible; i la modelització *especular* fa que la reutilització dels noms sigui possible a tot element del model (o quasi). Així, la *Modelització continua* és el resultat de la concurrència dels principis de l'*Espill* i de la *Franquícia obligada*.

Continuïtat entre la "realitat" i el model. En aplicar el principi de la *Modelització continua* obtenim que el model i la "realitat" modelitzada tenen estructures similars (o això hem intentat), i pels elements que es corresponen (un és la representació de l'altre) usem un mateix nom. D'aquesta manera la modelització no "trenca" ni crea una fissura en el nostre coneixement de les coses.

Traçabilitat de la modelització. Aquesta continuïtat entre el model i la "realitat" es veu reforçada amb l'aplicació del *Referent únic*: donat un nivell de modelització qualsevol, podem arribar a la "realitat" modelitzada per un nom del model seguint un *únic camí* que segueix la traça d'aquest nom a través de tots els nivells de modelització. Aquest camí únic és el que anomenem la *traça de la modelització* d'aquest nom.

Exemple 4 (Models en el desenvolupament)

La classe *Client* és la representació a nivell d'implementació del component *Client*, que ha aparegut en el disseny com a representació software del concepte *Client* que apareix en l'especificació del problema; el qual, al seu torn, és la manera d'expressar la idea de *Client* que tenen els usuaris del sistema.

La traça de la modelització de *Client* és la seqüència (única) que passa per una idea, un concepte, un component i una classe d'un llenguatge OOP, tots ells amb el nom *Client* (i no hi ha cap nom *Client* que no estigui dins d'aquesta traça).

2 Principis de desenvolupament de software

Es tracta de principis usats en les diferents etapes del desenvolupament d'un sistema software. La diferència amb els principis de modelització és que ara els models usats es corresponen a conceptes del software.

2.1 Encapsulament

Encapsulament. Cada unitat de la solució software té uns límits ben definits: sabem on comença, on acaba, i amb qui interactua.

Principi



Exemple 5 (Goto) En BASIC les subrutines només es diferencien perquè en el seu fil d'execució troben un `return`, i la seva crida es fa amb un `gosub` enlloc d'un `goto`. Imaginem un codi on sembla que hi ha una rutina A de la línia 100 a la 200 (tenim un `return` a la línia 200); i una altra subrutina B entre les línies 300 i 400. Suposem que a la línia 350 hi ha un `goto 180` i que en la línia 10 fem `gosub 320`. Llavors:

- Hem entrat a B per un punt intermedi, no pas per la seva primera línia
- A la línia 350 entrem a A, però no pas com a subrutina
- A la línia 200 trobem el `return` que en aquest cas ens fa el retorn de la crida de la línia 10

Llavors, quins són els límits de la subrutina B? I els de A? En BASIC els límits són inexistents; no hi ha encapsulament.

2.2 Ocultació

Ocultació. Cada unitat de la solució software oculta aquells detalls de la seva construcció que són susceptibles de canvis.

Principi



Dues perspectives: ús i construcció. El principi de la *Ocultació* introdueix la idea que tota unitat de la solució software es pot veure des de dues perspectives diferents: la perspectiva de l'ús i la perspectiva de la *construcció*.

Dos actors: client i servidor. Les unitats de la solució software s'usen executant-les d'alguna manera, per exemple, mitjançant una *crida* realitzada des d'una altra unitat de la solució software. En usar una unitat estem demanant que ens ofereixi un servei; l'usuari és el *client* d'aquest servei; la unitat usada és el *servidor* del servei.

Perspectiva i coneixement. El principi d'*Ocultació* ens diu que el coneixement que de la unitat software en té un client, és diferent de la que en té el servidor (això és, la pròpia unitat software usada). Com a servidors tenim tot el coneixement possible; com a clients hi ha informació que ens queda amagada, inaccessible.

Afectació local dels canvis. El principi d'*Ocultació* demana que allò que s'amagui sigui allò que és susceptible de canvi. D'aquesta manera els canvis produïts en el servidor (és a dir, en la construcció de la unitat software) no

afecten el client. Com a conseqüència, els canvis no es propaguen, sinó que només afecten localment.

2.3 Interfície

Definició. Interfície

Descripció de com usar una unitat de la solució software

En la interfície s'expliciten totes les maneres possibles d'usar la unitat de software. N'és el manual d'instruccions.

Exemple 6 En el món de la codificació C considerem les unitats software anomenades C-funcions. La interfície d'una C-funció ha d'expressar quin és el nom exacte d'aquesta (que serà el nom usat per fer les crides) i quina informació manipula. En concret caldrà donar la seqüència de tipus dels arguments necessaris, i la indicació de si la informació és entrant o sortint.

Cal observar com els noms (ficticis) dels paràmetres no formen part de la interfície (tot i que si que es poden incloure en la capçalera C dins el fitxer .h), ja que no donen cap mena d'informació sobre l'ús de la C-funció.

En el cas del C, la indicació de si la informació és entrant o sortint a la C-funció no és explícita. En alguns casos es pot extreure del tipus de l'argument; en d'altres de la presència de la paraula reservada `const`.

Principi



Interfície obligada. L'ús de cada unitat de la solució software està perfectament delimitat i explicitat.

El principi de la *Interfície obligada* el que ens diu és que tota unitat de la solució software té una *interfície* definida.

2.4 Contractes

Insuficiència de les interfícies. La preseència d'una *interfície* diu com es pot usar la unitat software, però no diu res dels contextos on es pot usar, ni de quins serveis són els que ofereix. Per a això necessitem els *contractes*.

Definició. Contracte

Descripció explícita del comportament i de l'ús (com usar i en quins contextos) una unitat de la solució software.

²No s'ha de confondre la *interfície* com a descripció explícita de com s'ha d'usar una unitat software amb conceptes homòlegs com per exemple les *interface* de llenguatges tipus Java, o els mecanismes d'interacció gràfica amb l'usuari extern del sistema (Interfície Gràfica d'Usuari, IGU). Potser un dels conceptes que més s'hi assembla del món de la programació i codificació són les capçaleres del C.

Manual del bon ús. En el contracte s'expliciten les situacions correctes d'ús de la unitat de software, i què és el que hom espera com a resultat de l'ús d'aquesta unitat. El contracte és el manual del bon ús i el manual del quan i el perquè ens cal l'ús.

Exemple 7 La C-funció *facturació*(*Empresa*) es preocupa de fer la facturació del client indicat amb l'argument de tipus *Empresa*. La *facturació()* només es pot realitzar dins la primera setmana de cada mes; i només per a aquells clients que tinguin albarans pendents de facturar. El resultat de llançar la petició és que al client en qüestió se li ha emès una factura per tots els albarans del mes anterior.

Hem respost les següents qüestions:

Què fa. Emetre una factura per tots els albarans del mes anterior del client indicat

Significat dels arguments. L'argument rebut ha de ser una *Empresa*, i representa el client al qui volem facturar

Contextos vàlids. Primera setmana de cada mes. El client sobre el que es vol facturar ha de tenir albarans pendents

Resultat. S'ha emès la factura amb els albarans pendents del mes anterior, els quals deixen d'estar pendents

Contractes PRE/POST. Un tipus de contracte molt usat és el que expressa els contextos en forma de *precondicions* (PRE) i *postcondicions* (POST).

Definició. Contracte PRE/POST

Contracte que s'expressa mitjançant els següents elements:

- Interfície
- Explicació semàntica del comportament i del significat de la informació manipulada per la interfície
- Condicions contextuais d'ús correcte (PRE)
- Condicions resultants d'un ús correcte (POST)

Lectura d'un contracte PRE/POST. La interfície diu com usar la unitat de software. La semàntica expressa quina és la funcionalitat d'aquesta unitat, i quin paper hi juga cadascun dels elements d'informació involucrats. Les condicions PRE i POST expressen els contextos d'ús correcte, i les condicions que s'obtenen sota un ús correcte.

Exemple 8 (Contracte PRE/POST)

Interfície. *facturacio (Empresa)*

Descripció. *Genera una factura amb els albarans del mes anterior del client indicat; i marca aquests albarans com a no pendents.*

Arguments. *Client sobre el que volem fer la facturació.*

PRE. *El client existeix dins el sistema i té albarans pendents del mes anterior. Estem dins la primera setmana del mes.*

POST. *S'ha generat la factura pels albarans pendents del mes anterior, corresponents al client; el client no té albarans pendents del mes anterior.*

2.5 Desenvolupament sota contracte

Principi



Desenvolupament sota contracte. El client només s'ha de preocupar d'assegurar les PRE; el servidor només s'ha de preocupar d'assegurar les POST.

Divisió de responsabilitats; confiança mútua. El principi del *Desenvolupament sota contracte* proposa una divisió de responsabilitats fonamental, basada en una *doble confiança*: el servidor ha de confiar en què el client només requereix els seus serveis sota els contextos apropiats; i al seu torn el client ha de confiar que el servidor, sempre i quan els serveis se li demanin en els contextos apropiats, es comportarà oferint el que li exigeix el contracte.

2.6 Caixa blanca i caixa negra

Definició. Caixa blanca

Unitat software encapsulada i amb interfície ben definida.

Origen de la nomenclatura. És una caixa perquè els límits estan ben perfilats (principi d'*Encapsulament*), i els usos estan explicitats en la *interfície* (només hi podem accedir per la tapa o interfície). És blanca³ perquè podem obrir la tapa i veure el seu interior, o alternativament mirar a través de les seves parets transparents.

Principi



Ocultació total. El client només coneix els *usos* de la unitat software.

³Seria millor dir que és *transparent*.

Coneixement limitat a la interfície. El principi de la *Ocultació total* diu que el client coneix la *interfície* de la unitat, però en desconeix les seves interioritats.

Definició. Caixa negra

Caixa blanca amb ocultació de tot el seu interior (l'únic visible és la interfície).

Principis que porten a la caixa negra. És a dir, una *caixa negra* és una unitat de software construïda segons els següents principis:

1. *Encapsulament*: La unitat està *encapsulada*
2. *Interfície obligada*: Els seus usos estan ben definits
3. *Ocultació total*: La *interfície* és l'únic que el client coneix de la unitat

Definició. Component

Caixa negra, amb contracte explícit, usada com a unitat d'assignació de responsabilitats en el disseny.

2.7 Desenvolupament amb caixes negres

2.7.1 Independència entre l'ús i la construcció

Dualitat client/servidor fins a l'extrem. L'ús de les *caixes negres* com a única unitat en el desenvolupament d'una solució software porta a l'extrem la doble perspectiva client/servidor.

Ús amb desconeixement. Els clients només coneixen la interfície de la caixa, i per tant tot desenvolupament en el que s'usi la caixa es pot fer (de fet no hi ha més remei que fer-ho així!!) sense saber com és per dins la caixa usada. L'ús es fa amb el total desconeixement de com s'ha fet la construcció.

Construcció aïllada. Al seu torn els servidors només s'han de preocupar de ser consistents amb la interfície que exposen: l'interior de la caixa que no s'exposi en la interfície no afecta l'ús que en fan els clients, pel simple fet que aquests no hi poden accedir. Per tant, la construcció es realitza de manera aïllada, sense tenir en compte on i com s'usa la caixa negra.

2.8 Desenvolupament sota contracte amb caixes negres

El contracte com a mecanisme de comunicació. El *desenvolupament sota contracte* amb *caixes negres* ens porta un pas més enllà. No només el client i el servidor es poden desenvolupar de manera independent sinó que a més la doble confiança que implica el desenvolupament sota contracte simplifica enormement els desenvolupaments. L'únic punt de contacte entre el client i el servidor, i que per tant cal mantenir estable, és el contracte ofert pel servidor.

El contracte *comunica* al client què s'espera d'ell quan usi la caixa negra; el contracte *comunica* al servidor què ha d'oferir al client sota les condicions adients.

Desenvolupament independent. Per desenvolupar el client no cal disposar del servidor, n'hi ha prou en usar-lo sota les condicions del seu contracte (és a dir, cal assegurar les PRE). Per desenvolupar un servidor no cal disposar de cap client; n'hi ha prou en assumir les condicions correctes d'ús (les PRE) i assegurar les condicions exigides pel contracte (les POST).

Algunes conseqüències. El desenvolupament sota contracte amb caixes negres permet fàcilment el desenvolupament descendent i el modular, entre d'altres: de les unitats o mòduls usats l'únic que cal explicitar és el contracte. També permet fàcilment la divisió de tasques: els contractes es converteixen en els punts de comunicació entre els diferents equips desenvolupadors.

3 Regles i principis

(2)

Els principis són recomanacions. Les regles obligacions.

En el disseny regnen els principis. En el disseny treballem amb principis. Per tant, en recomanacions que podem seguir o no, i que fins i tot en alguns casos poden esdevenir contradictòries.

Regles en el disseny. Alguns principis, però, els considerarem d'obligat compliment. Concretament, només considerem *caixes negres*; exigirem que aquestes explicitin un *contracte*; i usarem els contractes per dividir les responsabilitats del client i el servidor (principi de *Desenvolupament sota contracte*). És a dir, el disseny exigirà l'obligat compliment de les següents pautes i principis:

- Caixa negra com a únic tipus d'unitat
 - Encapsulament
 - Ocultació total
 - Interfície obligada
- Contracte explícit⁴
 - Desenvolupament sota contracte

⁴Usarem contracte PRE/POST

Principis i definicions del capítol

Principis

Desenvolupament sota
contracte, 14

Encapsulament, 11

Espill, 8

Franquícia, 9

Franquícia obligada, 9

Interfície obligada, 12

Modelització continua, 10

Ocultació, 11

Ocultació total, 14

Referent únic, 9

Definicions

Caixa blanca, 14

Caixa negra, 15

Component, 15

Contracte, 12

Contracte PRE/POST, 13

Interfície, 12

Model, 8

Capítol 2

Disseny: punt de partida i objectius

1 El punt de partida del disseny	22
1.1 Què ofereix l'especificació	22
1.2 Lectura dels contractes dels ES	22
2 Objectiu bàsic del disseny	23
3 Les tasques del dissenyador	24
3.1 Descripció de les tasques	24
3.2 Descomposició inicial	24
3.3 Recomposició de la descomposició inicial	24
3.4 El disseny més enllà de l'especificació	24
4 Assignació de responsabilitats	25
4.1 Com assignar una responsabilitat	25
4.2 Reconsideració d'una assignació	26
4.3 Una densa i fràgil teranyina de decisions	27
Principis	29

Contingut detallat del capítol 2

1 El punt de partida del disseny	22
1.1 Què ofereix l'especificació	22
El que ofereix l'especificació	22
Model conceptual (MC).	22
Cas d'ús (CU).	22
Esdeveniment de sistema (ES).	22
Precondicions (PRE).	22
Postcondicions (POST).	22
Contracte d'un ES.	22
1.2 Lectura dels contractes dels ES	22
ES de l'especificació i ES del disseny	22
Contracte dels ES del disseny	22
Contracte dels ES: doble naturalesa	23
Els contracte dels ES des de l'especificació	23
Els contracte dels ES des del disseny	23
Correspondència entre les dues visions	23
2 Objectiu bàsic del disseny	23
3 Les tasques del dissenyador	24
3.1 Descripció de les tasques	24
3.2 Descomposició inicial	24
Descomposició inicial i ES	24
3.3 Recomposició de la descomposició inicial	24
3.4 El disseny més enllà de l'especificació	24
Tasques pendents	24
Completesa en el disseny	24
4 Assignació de responsabilitats	25
4.1 Com assignar una responsabilitat	25
Metaprinicipis per a l'assignació de responsabilitats	25
Simultaneïtat	25
La teoria del tot o res	25
Igualtat d'oportunitats	25
Principis generals i específics	25
Recomanacions contradictòries	25
4.2 Reconsideració d'una assignació	26
Decisions des del coneixement parcial	26
Reconsideració contínua	26
Som la nostra història	27
Reconsideració	27
4.3 Una densa i fràgil teranyina de decisions	27
Un continu de decisions	27
Dos àmbits: el d'una decisió i el d'anàlisi per a la presa de la decisió	27
Risc de desastre	27

Teranyina	28
Xarxa de decisions	28
Convenciment i cautela	28
Principis	29

1 El punt de partida del disseny

1.1 Què ofereix l'especificació

El que ofereix l'especificació. L'especificació ens dona un *model conceptual* (MC) i els *casos d'ús* (CU), expressats en termes d'*esdeveniments de sistema* (ES). Per cada esdeveniment de sistema tenim el seu *contracte*: nom de l'ES, nom i significat dels paràmetres, *precondicions* (PRE) i *postcondicions* (POST).

Model conceptual (MC). Mostra els conceptes del domini i les interrelacions entre aquests. Les interrelacions principals són les associacions i les especialitzacions/generalitzacions.

El MC s'acostuma a presentar gràficament a través d'algun tipus de *Diagrama conceptual*. El diagrama per si sol, però, no constitueix el MC: hi ha molts aspectes del MC que no són representables gràficament.¹

Cas d'ús (CU). Ús prototípic del sistema que fa que aquest (el sistema) tingui un valor afegit per l'usuari.

Esdeveniment de sistema (ES). Petició que l'usuari fa al sistema, per sota de la capa de presentació.

Cada ES té una responsabilitat assignada; la combinació adient de les responsabilitats dels diferents ES permet exercir les responsabilitats dels CU. Aquesta combinació sovint es dona en forma de seqüència d'ES.²

Precondicions (PRE). Condicions que ha d'assegurar el "client". El "servidor" no les ha de comprovar pas.

Postcondicions (POST). Condicions que ha d'assegurar el "servidor".

Contracte d'un ES. Diu quines són les responsabilitats de l'ES: ha d'aconseguir les POST a partir de les condicions de les PRE.

1.2 Lectura dels contractes dels ES

ES de l'especificació i ES del disseny. Segons el principi de la *Modelització continua*,^{3, →} cada ES es modelitza amb un *esdeveniment de sistema software*.⁴ Així podem parlar dels *ES de l'especificació* i dels *ES del disseny*. Sovint parlarem simplement d'ES i deixarem que el context desambigui el terme.

¹I això és independent del llenguatge de modelització emprat. A vegades hi ha aspectes que el llenguatge de modelització permet expressar gràficament, però amb el cost de perdre simplicitat o claredat; en aquests casos sempre és preferible mantenir el diagrama simple, i afegir el que calgui textualment o d'alguna altra manera. En el cas de l'UML s'aplica la regla 80-20: un 80% del model és gràfic; l'altre 20% és text.

²En UML s'acostuma a usar un *diagrama de seqüència*.

³De fet només cal aplicar el principi de l'*Espill*. La referència a la *Modelització continua* ens diu que a més usarem els mateixos noms.

⁴Sovint es parla d'*operacions*, o fins i tot de *mètodes*. Aquí no utilitzem aquests termes perquè es poden confondre fàcilment amb termes específics de determinades tecnologies o determinats llenguatges de programació.

Contracte dels ES del disseny. El contracte d'un *ES del disseny* serà el correlat del contracte de l'*ES de l'especificació* corresponent. Segons el principi de la *Modelització continua* el contracte tindrà el mateix "aspecte" que el de l'especificació, però on allà es feia referència a una "realitat", aquí es fa referència a un model d'aquesta "realitat".

Contracte dels ES: doble naturalesa. Així podem veure els contractes dels ES com un sol artefacte amb dues lectures diferents: la lectura des de l'especificació i la lectura des del disseny.

Els contracte dels ES des de l'especificació. Com a elements de l'especificació, els contractes dels ES s'han de llegir en el *domini del problema*. En concret, per exemple, parlem de *conceptes*.

Els contracte dels ES des del disseny. Com a elements del disseny, els contractes dels ES s'han de llegir en termes del *domini de la solució* software que cal construir. En concret, per exemple, cada *concepte* que apareixia en la lectura des de l'especificació, en la lectura des del disseny caldrà llegir-lo com a *component*.

Correspondència entre les dues visions. En virtut de la *Modelització continua* cada element del domini de la solució que apareix en el contracte (visió des del disseny) és el correlat d'un element del mateix nom en el domini del problema (visió des de l'especificació).

Exemple 9 (ES de l'especificació i ES del disseny) *Sigui l'ES novaInscripcio(c:Caminada); en l'especificació el seu contracte exigeix que se li envii en forma d'argument la Caminada a la que fa referència la nova Inscripció; i la POST demana la creació d'una nova realització de i:Inscripció.*

Des del disseny el contracte diu que cal construir un ES (esdeveniment de sistema del software) que sigui el correlat de modelització de l'ES novaInscripcio(c:Caminada); que com a argument aquest ES ha de rebre una realització del correlat de disseny del concepte Caminada (és a dir, ha de rebre una realització del component Caminada); i que la POST de l'ES dissenyat exigeix que s'hagi creat una nova realització i d'un component Inscripció.

2 Objectiu bàsic del disseny

L'objectiu bàsic del disseny és determinar un conjunt de *components* \rightarrow col·laboratius tals que:

1. El seu comportament dinàmic fa el que demana l'especificació⁵, \rightarrow

⁵En el capítol 3. *Elements del disseny*, pàgina 31, veurem com els components són una abstracció que encapsula els comportaments possibles; i que per tant són les realitzacions dels components les qui de fet poden col·laborar. Tot plegat portarà a la revisió de la descripció de l'objectiu bàsic del disseny que acabem de presentar.

2. La implementació dels components és “fàcil”. És a dir, el disseny resol tots els aspectes de responsabilitats o col·laboracions, i deixa a la implementació la tasca de resoldre els aspectes tecnològics o del llenguatge de programació

3 Les tasques del dissenyador

3.1 Descripció de les tasques

Per aconseguir un bon disseny cal realitzar les següents tasques:

Descomposició. Descomposició de les tasques especificades en *responsabilitats* diferents i ben delimitades

Assignació. Assignar cadascuna de les responsabilitats al component adient⁶, →

Recomposició col·laborativa. Cal demanar als components que col·laborin convenientment, a través de l'enviament de *missatges*, per tal d'aconseguir el resultat desitjat

Definició pàgina 36

3.2 Descomposició inicial

Descomposició inicial i ES. El *comportament dinàmic* del sistema que volem dissenyar ens ve donat pels CU de l'especificació. En presentar els CU en termes d'ES de l'especificació ja se'ns dona la *descomposició inicial*.

3.3 Recomposició de la descomposició inicial

Aquesta també ens ve donada per l'especificació quan ens expressa (amb un *diagrama de seqüència*, per exemple) els diferents CU com una seqüència d'ES.

3.4 El disseny més enllà de l'especificació

Tasques pendents. L'especificació ofereix la descomposició i recomposició inicials, en termes d'ES. A partir d'aquí la tasca del dissenyador és la de dissenyar els ES especificats: descompondre'ls en responsabilitats, assignar-les a qui correspongui, i fer la recomposició col·laborativa pertinent.

⁶Pròpiament les responsabilitats s'assignen a objectes (un *objecte* és la realització d'un component; vegeu la definició a l'apartat 3.1.1.2., pàgina 36); però en dissenyar ens fixem més en les propietats d'aquests objectes que no pas en l'objecte concret. És a dir, els raonaments sobre a qui li assignem tal o qual responsabilitat els fem en termes de components, no pas d'objectes.

Quan en el mateix DC tinguem més d'una realització del mateix component (és a dir, dos “objectes” del mateix “tipus”) tindrà sentit raonar en termes d'objecte i no pas de components.

Completesa en el disseny. Hem de dissenyar tots els ES de tots els CU. L'ordre és indiferent.⁷

4 Assignació de responsabilitats

4.1 Com assignar una responsabilitat

Metaprinicipis per a l'assignació de responsabilitats. Per tal d'assignar les responsabilitats usem els principis de disseny. L'assignació d'una responsabilitat exigeix prendre una decisió: a quin component assignem la responsabilitat en qüestió. Tot seguit presentem els *metaprinicipis* que ens guien en aquesta presa de decisions.

Simultaneïtat. En cada decisió d'assignació de responsabilitat hi inter-venen tots els principis aplicables.

Principi



La teoria del tot o res.⁸ La *Simultaneïtat* diu que en el moment d'assignar una responsabilitat s'han de tenir en compte *tots* els principis que es puguin aplicar en la situació analitzada.

Igualtat d'oportunitats. Tots els principis participen en igualtat d'oportunitats: no hi ha cap principi que les seves recomanacions siguin "més fiables" o que predominin per sobre les recomanacions dels altres principis aplicats.

Principi



Principis generals i específics. El principi de la *Igualtat d'oportunitats* diu que davant determinades situacions específiques, la possibilitat d'aplicar principis dissenyats expressament per a aquestes situacions no ens eximeix de la necessitat d'aplicar altres principis aplicables en situacions més generals. *L'especificitat manté vàlida la generalitat.*

Recomanacions contradictòries. Els principis de la *Simultaneïtat* i de la *Igualtat d'oportunitats* deixen oberta la porta a la possibilitat que en el moment de prendre una decisió d'assignació de responsabilitat els diferents principis aplicats donin recomanacions diferents, àdhuc contradictòries. En aquests casos caldrà aplicar alguns dels següents criteris:

- **Quantitatiu.** Quina de les recomanacions ve avalada per més principis.

⁷Hi ha tècniques per decidir l'ordre, però no ens interessen de moment.

⁸Evidentment el disseny es basa en el "tot"; el "res" correspon al desenvolupament a la brava, sense disseny.

- *Principis avaluatius*. Hi ha principis que enlloc de donar una recomanació avaluen les recomanacions donades pels altres principis. L'aplicació dels principis avaluatius, però, també pot portar a contradiccions.⁹
- *Simetria*. Prendrem la mateixa decisió que hem pres anteriorment en una situació similar.¹⁰

4.2 Reconsideració d'una assignació

Decisions des del coneixement parcial. En el moment de prendre una decisió d'assignació de responsabilitat fem entrar en joc tot el coneixement del sistema que tenim en aquell moment. Aquest coneixement però és sempre parcial: fins que no haguem dissenyat tot el sistema no el coneixem en la seva totalitat. Per tant cada decisió d'assignació de responsabilitat es basa en el *coneixement parcial* que tenim en el moment de prendre la decisió.

Reconsideració contínua. Els principis permeten decidir quina és la millor decisió d'assignació de responsabilitat sota el coneixement present del sistema. Però això no vol dir que, necessàriament, aquesta fos la decisió que prendríem en cas de conèixer tot el sistema. Per tant, a mesura que anem dissenyant caldrà revalidar contínuament les decisions preses en el passat.

Exemple 10 (Localitat i reconsideració) *Suposem que el principi X diu que en casos de dissenys alternatius cal escollir aquell que assigni menys responsabilitats a cada component.¹¹ Compte: el principi recomana escollir el model tal que el nombre de responsabilitats de cada component sigui mínim; el que no diu és que la responsabilitat cal assignar-la al component que en aquests moments tingui menys responsabilitats assignades.*

Suposem que el disseny desenvolupat fins el moment conté dos components, A i B, el primer amb una responsabilitat assignada, i el segon amb dues. Suposem que els principis de disseny aconsellen assignar la responsabilitat R o bé a A o bé a B.

L'aplicació del principi X en aquesta situació ens porta a preferir assignar R a A: obtenim un disseny amb dues responsabilitats assignades a cada component; en cas d'haver assignat R a B obtindríem un disseny amb un component amb tres responsabilitats assignades.

Cal observar que estem aplicant el principi X al model parcial que tenim, no pas a tot el model. De fet estem assumint que si minimitzem el nombre de responsabilitats dels components en aquest model parcial, estarem minimitzant el nombre de responsabilitats dels components en el model total. Però això no necessàriament ha de ser així.

Suposem que a continuació els principis demanen assignar les responsabilitats R2 i R3 a A. El resultat és que el disseny assigna quatre respon-

⁹Els principis avaluatius també són principis perquè l'avaluació que donen és subjectiva. D'algun manera "recomanen" que ens decanem més per una recomanació o una altra.

¹⁰El criteri de la *Simetria* no és un criteri local a la responsabilitat que intentem assignar, sinó global a tot el sistema que estem dissenyant.

2.4 Assignació de responsabilitats

J.M. Merenciano

sabilitats a A i dues responsabilitats a B . Si per contra abans haguéssim pres la decisió d'assignar R a B ara obtindríem un disseny amb tres responsabilitats a cada component.

En termes del principi X aquest darrer disseny és preferible. La decisió presa en la situació inicial (correcta segons el coneixement que teníem del sistema) no és la millor des de la perspectiva global del sistema.

Som la nostra història. Un cop presa una decisió d'assignació de responsabilitat aquesta hipoteca el futur; i al seu torn, la decisió presa ve condicionada per les decisions preses en el passat. Això és perquè en tot moment les noves decisions d'assignació de responsabilitat han de ser consistents amb les decisions preses anteriorment.

Tot l'anterior queda resumit en el següent principi:

Reconsideració. Les decisions d'assignació preses s'han de reconsiderar contínuament. En concret, cada cop que, per mor de la presa de noves decisions d'assignació, canvien les condicions d'anàlisi que es consideraren en prendre-les.

Principi



4.3 Una densa i fràgil teranyina de decisions

Un continu de decisions. El disseny és un continu de preses de decisió. Per prendre una decisió cal analitzar tots els principis possibles (principi de *Simultaneïtat*) i les seves propostes (principi d'*Igualtat d'oportunitats*), i potser reconsiderar decisions anteriors (principi de la *Reconsideració*).

Dos àmbits: el d'una decisió i el d'anàlisi per a la presa de la decisió. Les decisions de disseny es preocupen de decidir a qui assignar determinada responsabilitat. És a dir, l'àmbit de cada decisió és la responsabilitat que ha d'assignar. Malgrat això l'àmbit d'anàlisi per prendre la decisió d'assignació és tot el sistema; i aquest s'ha construït a base de prendre decisions d'assignació d'altres responsabilitats. Per tant, tot i que l'àmbit d'una decisió és la responsabilitat que es vol assignar, l'àmbit d'anàlisi és el de totes les responsabilitats del sistema.

Risc de desastre. L'anàlisi feta davant una parcialitat d'informació s'ha de revalidar davant la globalitat del sistema. Cal assegurar que cada nova decisió presa no invalida tot el disseny desenvolupat fins el moment: de fet, en tot moment hi ha el risc de fer un pas en fals i que tot el disseny s'ensorri com un castell de cartes. La correcta aplicació dels principis (i metaprinicipis) de disseny minimitza al màxim aquest risc.

¹¹El principi de l'*Alta Cohesió* es pot llegir en termes similars.

Principi



Teranyina. En tot moment del desenvolupament, el disseny es pot veure com la xarxa (*Teranyina*) d'interrelacions entre les decisions d'assignació de responsabilitat que s'han pres per arribar al model present.

Xarxa de decisions. Les decisions no són independents. Venen condicionades pel passat i hipotequen el futur. El lligam entre les decisions diferents es poden expressar en una xarxa. A cada node de la xarxa hi figura la responsabilitat que es vol assignar, els principis aplicats, el coneixement usat per aplicar el principi, i l'assignació finalment presa.

Convenciment i cautela. En intentar estendre la xarxa amb un node més, pot ser que es trenquin lligams anteriors, i que haguem de reconfigurar-la (principi de la *Reconsideració*). En conseqüència, cap de les decisions que prenguem és inòcua. Tota decisió l'hem de prendre amb *convenciment* (justificat amb els principis de disseny) i amb *cautela* (comprovació que la reconsideració no és necessària).

Principis i definicions del capítol

Principis

Igualtat d'oportunitats, 25

Reconsideració, 27

Simultaneïtat, 25

Teranyina, 28

Capítol 3

Elements del disseny

1 Elements estructurals	36
1.1 Components i objectes	36
1.2 Responsabilitats	37
1.3 Missatges	39
1.4 Enllaços dirigits	40
2 Àmbit de validesa dels enllaços dirigits	44
2.1 Enllaç dirigit d'atribut	45
2.2 Enllaç dirigit local	46
2.3 Enllaç dirigit de paràmetre	48
2.4 La metàfora de la propietat oculta	51
3 Visibilitat	52
3.1 Visibilitat entre objectes	52
3.2 Visibilitat entre components	54
3.3 Multiplicitat de la visibilitat	57
3.4 Visibilitat indirecta	62
3.5 Delegació	68
3.6 Conclusions sobre la visibilitat	72
4 Col·laboració	73
5 Modelització de les associacions del MC	75
5.1 Les associacions en el MC	75
5.2 Model de les associacions en el disseny	75
Definicions	78

Contingut detallat del capítol 3

1 Elements estructurals	36
1.1 Components i objectes	36
1.1.1 Components	36
El <i>concepte</i> com a unitat bàsica de l'especificació	36
Unitat d'assignació de responsabilitats	36
Concepte i component	36
1.1.2 Objectes	36
Realitzacions dels components	36
1.1.3 Comportament	37
Encapsulació del comportament	37
Comportament dels objectes	37
Comportament: Capacitat i realització	37
1.2 Responsabilitats	37
1.2.1 Assignació de responsabilitats	37
Responsabilitat, missatge i operació	37
Assignació de responsabilitat a un component	37
Assignació de responsabilitat a un objecte	37
Exercici de les responsabilitats assignades	37
Component i objecte: dues cares d'una mateixa moneda	38
Dos nivells d'assignació	38
Dependència dels nivells d'assignació	38
1.2.2 Relectura de l'objectiu bàsic del disseny	38
Correcció en el subjecte de les col·laboracions	38
La paradoxa col·laborativa	38
Dos nivells; dues tasques	39
Relectura de l'objectiu bàsic del disseny	39
Assignació i recomposició	39
1.3 Missatges	39
Utilitat dels missatges	39
Actors d'un missatge	39
Mecanisme dels missatges	39
1.4 Enllaços dirigits	40
1.4.1 Canal dels missatges	40
L'enllaç dirigit com a canal	40
Múltiples enllaços emergents	40
Múltiples enllaços incidents	40
Semàntica dels enllaços dirigits	42
1.4.2 Context d'emissió dels missatges	42
El context d'emissió com a realització	42
Context d'emissió i origen de l'enllaç	43
Missatge i destinació de l'enllaç	43
Condicions d'emissió de missatges	43
2 Àmbit de validesa dels enllaços dirigits	44

	Validesa dels enllaços dirigits	44
	Àmbit de validesa dels enllaços dirigits	44
2.1	Enllaç dirigit d'atribut	45
	Independència del context d'emissió	45
	Múltiples usos per un sol enllaç	45
	Excursió. (Implementació)	45
2.2	Enllaç dirigit local	46
	Un enllaç dirigit per cada realització	46
	Context de localitat i context d'emissió	46
	Dependència del context d'emissió	46
	Un enllaç per cada ús	46
	Ocultació de l'enllaç	47
	Excursió. (Implementació)	47
2.3	Enllaç dirigit de paràmetre	48
	Publicació de la destinació	48
	Excursió. (Implementació)	48
	Semàntica de la publicació via argument	49
	Semàntica de la publicació via retorn	50
2.4	La metàfora de la propietat oculta	51
	Excursió. (Mecanisme de pas de paràmetres)	51
3	Visibilitat	52
3.1	Visibilitat entre objectes	52
	Visibilitat, enllaços i àmbits	53
	Visibilitat estable	53
	Visibilitat variable	53
	Visibilitat i existència	53
	Excursió. (Altres tipus de visibilitat)	54
3.2	Visibilitat entre components	54
3.2.1	La visibilitat com a potencialitat	54
	Potencialitat de col·laboració	54
	Encapsulació de la visibilitat en el component	54
	Visibilitat entre components en termes de la visibilitat d'objectes	55
	No obligatorietat de la realització de la visibilitat	55
	Obligatorietat de la realització de la visibilitat	55
	Enllaç dirigit i visibilitat de component: dues cares d'una mateixa moneda	55
	Excursió. (Visibilitat obligada i visibilitat optativa)	55
3.2.2	La visibilitat com a abstracció	56
	Enllaços dirigits com a realització de la visibilitat	56
	Visibilitat com a abstracció dels enllaços dirigits	56
	La visibilitat i l'abstracció dels enllaços dirigits	56
3.2.3	Àmbits de visibilitat	56
	Visibilitat, enllaços i àmbits	57
	Potencialitats de la capacitat d'atribut	57
	Potencialitats de la capacitat local	57
	Potencialitats de la capacitat de paràmetre	57

3.3	Multiplicitat de la visibilitat	57
3.3.1	Multiplicitat dels enllaços dirigits	57
	Múltiples destinacions	57
	Restricció dels enllaços dirigits multiavaluats	58
	Enllaç dirigit multiavaluat versus múltiples enllaços simples	58
	Missatge sobre un enllaç dirigit multiavaluat	59
	Excursió. (Implementació)	59
	Generalització del concepte d' <i>enllaç dirigit</i>	59
3.3.2	Multiplicitat de la visibilitat d'objectes	60
	Excursió. (Visibilitat multiavaluada d'objecte)	60
	Visibilitat generalitzada (d'un objecte)	61
	Excursió. (Sobre la notació)	61
3.3.3	Multiplicitat de la visibilitat de components	61
	Excursió. (Visibilitat multiavaluada d'un component)	61
	Visibilitat generalitzada (d'un component)	62
3.4	Visibilitat indirecta	62
3.4.1	Indirecció i enllaços dirigits	62
	Generalització de l'enllaç dirigit	63
	Definició recursiva	63
	Definició iterativa	63
	Aspectes de la notació	63
3.4.2	Cadena d'enllaços dirigits	63
	La cadena de la indirecció	63
	Cadena d'enllaços dirigits i visibilitat	64
	Seqüència de visibilitats sense cadena d'enllaços dirigits	64
	Excursió. (Condicció suficient per a la continuïtat)	65
3.4.3	Indirecció i cadenes de visibilitat	65
	Aspectes de la notació	65
	Cadena de visibilitats	66
	Cadena de visibilitats i cadena d'enllaços dirigits	66
	Realització d'una visibilitat indirecta	66
	Col·laboració en una cadena de visibilitats	66
	Col·laboració a través de la visibilitat indirecta	67
3.4.4	Àmbit de la visibilitat indirecta	67
	Cadena homogènia d'enllaços dirigits	67
	Visibilitat homogènia	67
3.4.5	Multiplicitat de la visibilitat indirecta	68
	Cadena de visibilitats directes monoavaluades	68
3.5	Delegació	68
	El mecanisme de la delegació	68
	Una responsabilitat; diverses operacions	69
	Un nom; diferents operacions	69
	"Retransmissió" o "reenviament"	69
	Excursió. (Implementació)	69
	Visibilitat de delegació multiavaluada	70
	Visibilitat de delegació indirecta	71

Cadena de delegacions	71
3.6 Conclusions sobre la visibilitat	72
Potencialitat	72
Potencialitat realitzada	72
Enllaç dirigit	72
Visibilitat d'objectes	72
Una moneda amb dues cares	72
Multiavaluació dels enllaços	72
Multiavaluació de la visibilitat	72
Col·laboració i visibilitat multiavaluada	72
Indirecció en els enllaços dirigits	72
Visibilitat indirecta	72
Cadenes de visibilitat	73
Col·laboració i visibilitat indirecta	73
Sobre la notació	73
Col·laboració per delegació	73
4 Col·laboració	73
Tipus de col·laboracions	73
Col·laboració d'enllaç	73
Col·laboració amb selecció	73
Col·laboració amb el grup	74
5 Modelització de les associacions del MC	75
5.1 Les associacions en el MC	75
Semàntica de les associacions de MC	75
Estaticitat de les associacions del MC	75
5.2 Model de les associacions en el disseny	75
5.2.1 Associacions del MC i visibilitat d'atribut	75
Semàntica de la visibilitat d'atribut	75
Visibilitat d'atribut i associació	76
Model de les associacions	76
Semàntica dels enllaços dirigits	76
Múltiples models de les associacions	76
Excursió. (Implementació)	76
Direccionalitat dels enllaços	77
La responsabilitat en l'orientació dels enllaços	77
Definicions	78

1 Elements estructurals

1.1 Components i objectes

1.1.1 Components

El concepte com a unitat bàsica de l'especificació. En l'especificació la unitat bàsica és el *concepte*: el *model conceptual* (MC) mostra els conceptes del problema i les seves interrelacions; els *ES* manipulen i actuen sobre conceptes.

Unitat d'assignació de responsabilitats. La unitat bàsica del disseny és la *unitat d'assignació de responsabilitats*. Els principis bàsics del desenvolupament exigeixen que les unitats del disseny siguin *caixes negres*. D'aquí la \rightarrow definició de *component* de l'apartat 1.2.6. *Caixa blanca i caixa negra*, pàgina 14.

Concepte i component. En virtut dels principi de l'*Espill* la unitat bàsica del disseny ha de ser el correlat software d'un concepte. Per tant, pel principi de la *Franquicia obligada*, un *component* de nom X és el correlat software d'un *concepte* de nom X. És a dir, cada *component* el podem veure com el "model", en termes de solució software, del *concepte* del mateix nom que apareix en el problema.¹

(3)

Un *component* és l'abstracció software que modelitza un *concepte*

1.1.2 Objectes

Realitzacions dels components. Un *concepte* és una abstracció de totes les realitzacions amb les mateixes propietats. Per tant, un *component* també és una abstracció; les realitzacions dels components són el correlat software de les realitzacions dels conceptes.

Definició. Objecte

*Donem el nom d'objecte a la realització d'un component.*²

¹Si la idea de *component* és el model de la idea de *concepte*, pel principi de la *Franquicia obligada*, els *components* s'haurien d'anomenar *conceptes*. Malgrat tot aquí violem aquest principi per facilitar la ubicació dels raonaments en el nivell corresponent. De fet el que fem és aplicar el principi de la *Franquicia obligada* als elements de l'especificació i l'anàlisi concrets (*client*, *albarà*, *caminada* etc.), i no pas als metaelements del desenvolupament (*concepte*, *component*, *classe*, etc.).

²Els *objectes* com a realitzacions de components no s'han de confondre amb els conceptes homònims d'alguns llenguatges de programació; com per exemple els objectes dels llenguatges OOP, que es defineixen com a realitzacions de classes (i on *classe* és una estructura del llenguatge emprat).

1.1.3 Comportament

Encapsulació del comportament. Els *components* són de naturalesa distinta als *conceptes*, i no només pels mons diferents en els que es mouen (problema i solució software). Els components són *caixes negres*, i per tant encapsulen els possibles usos o comportaments de les seves realitzacions; aquests usos són els indicats en la interfície del component.

Comportament dels objectes. L'encapsulació del comportament en els components significa que les diferents possibilitats d'actuació dels objectes estan encapsulades en el component del qual en són una realització. És a dir, el possible comportament dels objectes queda delimitat i definit pels respectius components.

Comportament: Capacitat i realització. El *component* encapsula les capacitats. Els *objectes* són els qui realitzen aquestes capacitats. El comportament d'un objecte és la realització del comportament definit pel component corresponent.

El *component* encapsula els possibles comportaments dels *objectes* que en són realització

(4)

1.2 Responsabilitats

1.2.1 Assignació de responsabilitats

Responsabilitat, missatge i operació. Assignar una *responsabilitat* significa donar la capacitat d'exercir aquesta responsabilitat. Per demanar l'exercici de la responsabilitat assignada caldrà enviar la petició corresponent, en forma de *missatge*. Finalment, per respondre a un missatge qui rep la petició realitzarà una *operació* determinada. En virtut del principi del *Modelització continua*, responsabilitat, missatge (petició) i operació (resposta) rebran el mateix nom.

Notació. En el que segeuix sovint considerarem com a sinònims els termes *responsabilitat*, *missatge* i *operació*.



Assignació de responsabilitat a un component. Assignar una responsabilitat a un component significa que qualsevol de les seves realitzacions té la capacitat per exercir aquesta responsabilitat: recordem que el component encapsula el comportament possible de les seves realitzacions.

Assignació de responsabilitat a un objecte. Assignar una responsabilitat a un objecte vol dir que la capacitat d'exercir aquesta responsabilitat ha d'estar encapsulada en el component del qual aquest objecte n'és una realització. Per tant qui de fet rep l'assignació de la responsabilitat és el component, no pas l'objecte.

Exercici de les responsabilitats assignades. Els *components*, com a abstraccions que són, no poden exercir les responsabilitats: no poden realitzar cap operació en resposta a les peticions. Els qui capturen les peticions i realitzen les operacions pertinents són els *objectes*. Així, si bé els *components* són qui reben les assignacions de responsabilitat, són els *objectes* concrets qui les realitzen.

Component i objecte: dues cares d'una mateixa moneda. Les capacitats de les realitzacions s'encapsulen en els *components*, però aquests només tenen sentit si hi ha *objectes* que les realitzen. Per altra banda, un *objecte* només pot existir i actuar d'acord amb les capacitats del *component* del quan n'és una realització:

(5)

- Els *components* no poden existir sense *objectes* que els realitzin
- Els *objectes* només poden existir i comportar-se d'acord amb el *component* del quan en són una realització

Dos nivells d'assignació. En raonar sobre l'assignació de responsabilitats podem pensar en la capacitat d'exercir-les, i per tant raonem sobre els *components*; o bé podem pensar en l'exercici concret, i per tant raonem sobre els *objectes*.

(6)

- Les responsabilitats s'assignen als *components*
- Les responsabilitats les exerceixen els *objectes*

Dependència dels nivells d'assignació. Sigui quin sigui el nivell d'assignació en el que es treballi, sempre caldrà plasmar les capacitats en realitzacions (els *components* existeixen perquè hi ha *objectes* que els realitzen), i les realitzacions en capacitats (el comportament dels *objectes* ha de ser coherent amb les capacitats del *component* del quan en són una realització). És a dir, els dos nivells sempre hi han d'estar presents.

1.2.2 Relectura de l'objectiu bàsic del disseny

Correcció en el subjecte de les col·laboracions. L'objectiu bàsic del disseny és la construcció d'un conjunt d'*objectes* (i no pas de *components*)^{3,→} col·laboratius que es comporti de la manera que exigeix l'especificació: són els *objectes* els qui col·laboren fent-se peticions mútues d'exercici de responsabilitat, no pas els *components*.

³La construcció d'un conjunt de components col·laboratius és el que demanàvem a l'apartat 2.2.Objectiu bàsic del disseny, pàgina 23.

La paradoxa col·laborativa. Les possibles col·laboracions entre els diferents *objectes* queden definides en els *components*: són els *components* qui reben les assignacions de responsabilitat. Però són les realitzacions d'aquests qui les exerceixen.

Dos nivells; dues tasques. En la tasca d'*assignació de responsabilitats*[→] s'assigna una responsabilitat a un *component*; en la *recomposició col·laborativa* es fan les peticions pertinents per tal que s'exerceixin les responsabilitats assignades. L'*assignació de responsabilitats* és una tasca sobre els *components*; la *recomposició col·laborativa* ho és sobre els *objectes*: cal decidir quins objectes envien quines peticions, a qui, i quan.

2.3.Les tasques del dissenyador, pàgina 24

Relectura de l'objectiu bàsic del disseny. En el conjunt de components hi ha totes les responsabilitats definides i assignades, i per tant totes les possibles col·laboracions. El disseny però també ha de mostrar com col·laboren les diferents realitzacions d'aquests components per obtenir el comportament exigint per l'especificació.^{4, →} En aquest sentit la frase "*Conjunt de components col·laboratius que es comporten de la manera volguda*" cal entendre-la com: el "conjunt de components, amb indicació de les responsabilitats assignades; i conjunt d'objectes col·laboratius que realitzen aquestes responsabilitats".

3.1.2.1.Assignació de responsabilitats, pàgina 37

El disseny ha de mostrar les responsabilitats assignades (quines i a qui), i les ha d'exercir.

(7)

Assignació i recomposició. La col·laboració pròpiament dita es fa entre els objectes; però la potencialitat s'expressa en els components. Per tant pertoca a la tasca de la *recomposició col·laborativa*[→] analitzar els *enllaços dirigits*; i a la tasca de l'*assignació de responsabilitats* li pertoca l'anàlisi de les *visibilitats*. Ara bé, l'estret lligam entre els enllaços dirigits i les visibilitats gairebé exigeix realitzar ambdues tasques sinó simultàniament sí en paral·lel.

2.3.Les tasques del dissenyador, pàgina 24

1.3 Missatges

Utilitat dels missatges. La col·laboració entre els objectes es fa a través dels *missatges*.

Actors d'un missatge. Tot missatge té un *objecte emissor* i un *objecte receptor*.^{5, →}

3.3.3.Multiplicitat de la visibilitat, pàgina 57

⁴El disseny ha de resoldre els dos nivells d'assignació de responsabilitat explicats a l'apartat 3.1.2.1.Assignació de responsabilitats, pàgina 37.

⁵En el cas d'usar *enllaços dirigits multiavaluats* o *enllaços generalitzats* com a canal de transmissió del missatge, el que tenim és un *conjunt de receptors*. Aquesta multiplicitat del receptor s'introdueix a l'apartat 3.3.3.Multiplicitat de la visibilitat, pàgina 57

Mecanisme dels missatges. Quan un *emissor* envia un missatge a un *receptor*, el que fa és enviar al *receptor* una petició per tal que aquest exerceixi determinada *responsabilitat* de la seva competència. Com a resposta el *receptor* realitza l'*operació* que modelitza la responsabilitat demanada.

1.4 Enllaços dirigits

1.4.1 Canal dels missatges

L'enllaç dirigit com a canal. Els missatges s'envien a través dels *enllaços dirigits*: per tal d'enviar un missatge, l'objecte emissor ha de tenir un *enllaç dirigit* cap a l'objecte receptor; és sobre aquest enllaç que "circula" el missatge.

Definició. Enllaç dirigit

Diem que hi ha un enllaç dirigit^a entre l'objecte oe i l'objecte or quan:

- *oe té coneixement de l'existència de or*
- *oe té un mecanisme per accedir a or (és a dir, un mecanisme per fer-li peticions, per enviar-li missatges).*

Anomenem origen de l'enllaç a l'objecte oe , i destinació de l'enllaç a l'objecte or .

^aUsem oe per objecte emissor; i or per objecte receptor.



Notació. Per expressar un *enllaç dirigit* de oe a or escriurem $oe \rightarrow or$.

Múltiples enllaços emergents. Res impedeix que un objecte oe tingui més d'un *enllaç dirigit*, cadascun amb un objecte destinació possiblement diferent.

Múltiples enllaços incidents. Res impedeix que sobre un objecte or hi hagi més d'un *enllaç dirigit*, cadascun amb origen en un objecte possiblement diferent.

Exemple 11 (Enllaços dirigits (Java))

```
class X                      // JAVA
{
    void m()
    {
        A u, v;              // 1
        u = new A();         // 2
        v = u;                // 3
    }
}
```

3.1 Elements estructurals

J.M. Merenciano

A (1) es defineixen dos identificadors de variables.⁶ La seva funció és la de mantenir, cadascun d'ells, un enllaç dirigit. L'origen de l'enllaç serà l'objecte $oe : X$ considerat. La destinació de l'enllaç és un objecte de tipus A ; l'objecte concret al qual apunta l'enllaç dirigit depèn de l'assignació que fem sobre la variable.

A (2) creem un objecte anònim de tipus A i el convertim en la destinació de l'enllaç dirigit u : a partir d'aquest moment a través de la variable u podrem accedir a l'objecte anònim acabat de crear.

A (3) definim com a destinació de l'enllaç dirigit v el mateix objecte accessible des de u . Així l'objecte creat a (2) té dos enllaços dirigits que l'apunten: u i v .

Exemple 12 (Enllaços dirigits (C++)) Anem a repetir l'exemple anterior en llenguatge C++. Si usem una política d'assignació per còpia (que és una pràctica molt habitual), llavors les coses són força diferents. Per analitzar-ho comencem per adaptar el codi de l'exemple a l'operador `new` del C++.

```
class X                                // C++
{
    void m()
    {
        A u, v;                        // 1
        u = *(new A());                // 2
        v = u;                          // 3
    }
}
```

Ara a (2) es creen dos objectes: el que crea el `new`, i una còpia d'aquest, que crea l'operador d'assignació. La destinació de l'enllaç dirigit definit per la variable u és l'objecte creat per l'operador d'assignació, i no pas de l'objecte creat per l'operador `new`. (De fet l'objecte creat per l'operador `new` es converteix en un objecte zombie: existeix, i per tant ocupa memòria, però no tenim cap enllaç dirigit sobre ell; és un objecte mort, ja que no hi tenim accés, però que es comporta com un objecte viu pel que fa a recursos de la màquina. De fet l'aparició d'aquests objectes zombies són el que fan que molts dels programes en C++ donin error en temps d'execució per manca de memòria).

A (3) es crea un nou objecte, còpia de l'objecte accessible mitjançant u . Un cop creat, aquest nou objecte es defineix com a destinació de l'enllaç dirigit v . En resum, s'han creat tres objectes, un d'ells zombie, i els altres dos cadascun com a destinació d'un enllaç dirigit diferent.

⁶Com que encara desconeixem els elements del disseny, els primers exemples els fem en termes de la implementació. Entenem que el lector està avesat en algun llenguatge de programació, i per tant la discussió li és molt més propera. Cal però entendre que el que es vol exemplificar són conceptes del disseny i no pas de la implementació.

Semàntica dels enllaços dirigits. Un enllaç dirigit permet posar en comunicació diferents objectes;^{7,→} el motiu pel qual posem aquests objectes en contacte és el que podem anomenar la *semàntica* de l'enllaç dirigit.^{8,→}

Exemple 13 (Semàntica d'un enllaç dirigit) Tota etapa d'una caminada té dos indrets singulars: l'indret d'inici i l'indret de finalització de l'etapa. Suposem^{9, →} que el disseny requereix que l'objecte $e:Etapa$ ha d'enviar el missatge *visitat()* als seus indrets singulars cada cop que algun participant en la caminada passa pel control de final d'etapa.

Una manera de resoldre-ho seria assegurant que cada $e:Etapa$ tingui dos enllaços dirigits: $e:Etapa \rightarrow i:Indret$, on $i:Indret$ representa l'inici de l' $e:Etapa$; $i:e:Etapa \rightarrow f:Indret$, on $f:Indret$ representa l'indret de finalització de l' $e:Etapa$.

Així la semàntica de l'enllaç dirigit $e:Etapa \rightarrow i:Indret$ és la d'"indret d'inici de l'etapa", mentre que la semàntica que li correspon a l'enllaç dirigit $e:Etapa \rightarrow f:Indret$ és la d'"indret de finalització de l'etapa".

1.4.2 Context d'emissió dels missatges

Definició. Context d'emissió d'un missatge

Anomenem context d'emissió d'un missatge $m()$ a la realització de l'operació que s'està realitzant en el moment d'emetre el missatge $m()$

El context d'emissió com a realització. En la construcció o descripció d'una operació s'indiquen quins missatges s'han d'enviar, és a dir, es descriu una col·laboració. Però una operació no pot emetre missatges; són les realitzacions de les operacions qui els emeten. Són les realitzacions de les operacions qui realitzen la col·laboració descrita. És per això que el *context d'emissió és una realització d'una operació, i no pas una operació*.

⁷En cas d'un enllaç dirigit monoavaluat intervenen dos objectes, l'origen i la destinació de l'enllaç dirigit; en el cas d'un enllaç dirigit multiavaluat (que s'introdueix a l'apartat 3.3.3.1. Multiplicitat dels enllaços dirigits, pàgina 57) intervenen més de dos objectes: en concret l'objecte origen de l'enllaç dirigit i tots els objectes que són destinació d'aquest enllaç dirigit.

⁸A l'apartat 3.5. Modelització de les associacions del MC, pàgina 75, veiem que la visibilitat d'atribut és la modelització de les associacions de MC; i per tant els enllaços dirigits d'atribut són la modelització dels enllaços, previstos a MC, entre les realitzacions de conceptes. Per tant, la semàntica de l'enllaç dirigit d'atribut es correspon a la semàntica de de l'associació a la que pertany l'enllaç que modelitza.

⁹A l'exemple 22, pàgina 58, es presenta una solució de disseny alternativa.

Notació. Per tal de simplificar la redacció, sovint direm que $m()$ és un *context d'emissió* quan en realitat caldria dir que el *context d'emissió* és una realització de $m()$.



Context d'emissió i origen de l'enllaç. Sigui $m()$ el *context d'emissió* d'un missatge $m1()$.¹⁰ Evidentment l'*origen de l'enllaç* a través del qual s'envia $m1()$ ha de ser la realització d'un component que ofereixi l'operació $m()$ entre les seves operacions possibles.

El context d'emissió d'un missatge $m1()$ és la realització d'una operació que ofereix el component del qual l'*origen de l'enllaç* és una realització.

(8)

Missatge i destinació de l'enllaç. El fet d'enviar un missatge $m1()$ a través d'un enllaç dirigit exigeix que l'objecte *destinació de l'enllaç* disposi de l'operació per respondre el missatge (que pel principi de la *Modelització continua* també es dirà $m1()$).

Per tal de poder enviar un missatge $m1()$ a través d'un enllaç dirigit, cal que el component del qual la *destinació de l'enllaç* és una realització ofereixi l'operació $m1()$.

(9)

Condicions d'emissió de missatges. Des d'un punt de vista informal totes les observacions anteriors es poden expressar de la següent manera:

- Per poder enviar un missatge, l'emissor $a:A$ ha de tenir visibilitat del receptor $b:B$ del missatge; per tant l'emissor $a:A$ ha de ser l'origen de l'enllaç dirigit emprat
- Per poder rebre un missatge, el receptor $b:B$ ha de ser vist per l'emissor del missatge; per tant el receptor $b:B$ ha de ser la^{11,→} destinació de l'enllaç dirigit emprat
- Per poder enviar un missatge, l'emissor $a:A$ ha d'estar fent alguna cosa; en concret ha d'estar responant un missatge $m()$
- Si $a:A$ està responant el missatge $m()$ és perquè el component A ofereix l'operació $m()$

3.3.3.1. Multiplicitat dels enllaços dirigits, pàgina 57

¹⁰Pròpiament, el context d'emissió de $m1()$ és una realització de l'operació $m()$.

¹¹En el cas d'usar un enllaç dirigit multiavaluat (que s'introdueix a l'apartat 3.3.3.1. Multiplicitat dels enllaços dirigits, pàgina 57) el receptor $b:B$ ha de ser *una* de les destinacions de l'enllaç dirigit emprat.

- Si $b : B$ rep el missatge $m1()$ és perquè el component B ofereix l'operació $m1()$

Exemple 14 (Context d'emissió) *Sigui el següent codi Java:*

```
class A
{
    B b = new B();

    void m()
    {
        b.m1();           // ****
    }
}
```

El que diu aquest codi és que tota realització de la classe A té un enllaç dirigit cap a una realització de la classe B . És a dir, hi ha una visibilitat $A \rightarrow B$, que es materialitza en enllaços dirigits $a : A \rightarrow b : B$.

Sigui a una realització de A , i suposem que en algun moment el codi Java té la instrucció $a.m()$. És a dir, s'envia un missatge $m()$ a l'objecte a , per tal que aquest respongui executant l'operació $m()$.

La línia marcada amb asterics diu que, com a resposta a la recepció del missatge $m()$, l'objecte a envia el missatge $m1()$ a l'objecte b que és la destinació de l'enllaç dirigit $a : A \rightarrow b : B$.

El context d'emissió de $m1()$ és (una realització de) $m()$. L'origen de l'enllaç dirigit a través del qual s'envia el missatge $m1()$ és un objecte d'una classe que ofereix l'operació $m()$. La destinació de l'enllaç dirigit a través del qual s'envia el missatge $m1()$ és un objecte d'una classe que ofereix l'operació $m1()$.

2 Àmbit de validesa dels enllaços dirigits

Validesa dels enllaços dirigits. Els enllaços dirigits es poden crear i destruir. Per tant cal assegurar que en el moment d'enviar un missatge l'enllaç dirigit pertinent és vàlid.

Àmbit de validesa dels enllaços dirigits. A continuació es presenta una classificació dels enllaços dirigits en funció del seu *àmbit de validesa*, és a dir, del context en el qual l'enllaç manté la seva validesa.

2.1 Enllaç dirigit d'atribut

Definició. Enllaç dirigit d'atribut

Direm que un enllaç dirigit és d'atribut quan la vida de l'enllaç coincideix amb la de l'objecte origen d'aquest enllaç.

Independència del context d'emissió. Un *enllaç dirigit d'atribut* és aquell que és vàlid independentment tant de l'operació que s'estigui realitzant, com de la realització concreta d'aquesta operació. Per tant és independent del context d'emissió $m()$. És a dir, mentre mantinguem el mateix origen de l'enllaç dirigit, tots els contextos d'emissió vàlids (això és, tota realització d'alguna de les operacions que pot realitzar l'objecte que és origen de l'enllaç) comparteixen el mateix enllaç dirigit, i per tant el mateix objecte destinació de l'enllaç dirigit.

Múltiples usos per un sol enllaç. L'*enllaç dirigit d'atribut* permet que un mateix enllaç dirigit sigui reutilitzat en diferents usos: cada nova realització d'una mateixa operació, així com de diferents operacions, comparteixen el mateix enllaç dirigit.

Excursió. (Implementació) En termes de programació OO, per exemple, l'objecte oe origen de l'enllaç dirigit és d'una classe C_e que conté un membre de dades (o atribut) de nom or que és un objecte d'una classe C_r .

D'aquesta manera, cada objecte $oe:C_e$ té el seu propi enllaç dirigit (el valor de l'atribut $or:C_r$); i durant l'execució de qualsevol mètode $m()$ per part de l'objecte $oe:C_e$ es podrà accedir al mateix objecte destinació $or:C_r$ (que és potencialment diferent per cada objecte emissor $oe:C_e$).

Exemple 15 (Enllaç dirigit d'atribut)

```
class A
{
    B b = new B();

    void m1()
    {
        b.f();
    }

    void m2()
    {
        b.f();
    }
}
```

Donat un objecte $a:A$ el missatge $f()$ té com a contextos d'emissió possibles qualsevol realització de l'operació $m1()$ i qualsevol realització de l'operació $m2()$.

Donat un objecte $a : A$ existeix un enllaç dirigit (d'atribut) $a : A \rightarrow b : B$ que és utilitzable tant per l'operació $m1()$ com per l'operació $m2()$. Així $a.m1()$ i $a.m2()$ usen el mateix enllaç dirigit $a : A \rightarrow b : B$.

El codi següent mostra dos contextos d'emissió diferents pel missatge $f()$ però que són realitzacions diferents d'una mateixa operació. L'enllaç $a : A \rightarrow b : B$ emprat és el mateix en ambdós contextos d'emissió.

```
A a = new A();  
a.m1();           // 1  
a.m1();           // 2
```

2.2 Enllaç dirigit local

Definició. Enllaç dirigit local

Direm que un enllaç dirigit és local ($a.m()$) quan la validesa de l'enllaç dirigit es limita a una realització de l'operació $m()$.

Definició. Context de localitat

Donat un enllaç dirigit local, anomenem context de localitat d'aquest enllaç la realització concreta de l'operació $m()$ dins la qual l'enllaç dirigit té validesa.

Un enllaç dirigit per cada realització. Un *enllaç dirigit local* és un enllaç dirigit diferent per cada operació i per cada nova realització d'una mateixa operació $m()$; és a dir, és un enllaç dirigit diferent per cada *context de localitat*.

Context de localitat i context d'emissió. El *context d'emissió* d'un missatge enviat a través d'un enllaç dirigit local és necessàriament el *context de localitat* d'aquest enllaç. Això és així perquè l'enllaç dirigit només existeix o és vàlid dins del seu context de localitat, i per tant tot ús correcte de l'enllaç dirigit ha de ser dins d'aquest context.

Dependència del context d'emissió. La coincidència del context de localitat i del context de d'emissió en cada ús de l'enllaç dirigit local permet afirmar que la validesa d'un enllaç dirigit local depèn del context d'emissió. És a dir, no només depèn de quina operació s'està realitzant en el moment d'emetre el missatge, sinó que depèn de cada realització concreta d'aquesta operació.

Un enllaç per cada ús. L'*enllaç dirigit local* és vàlid per un sol ús d'una operació $m()$: cada nou ús, cada nova realització de $m()$ és un nou *context de localitat*, i per tant es crea un enllaç dirigit local nou.

3.2 Àmbit de validesa dels enllaços dirigits

J.M. Merenciano

Notació. Per tal de simplificar la redacció, sovint direm que $m()$ és un *context de localitat* quan en realitat caldria dir que el *context de localitat* és una realització de $m()$.



Ocultació de l'enllaç. L'enllaç dirigit usat en un *context de localitat* $m()$ deixa de ser vàlid o accessible un cop acabada la tasca de $m()$.¹² Per tant, l'*enllaç dirigit local* (i especialment la seva destinació) queda ocult dins del context de localitat $m()$.¹³

Excursió. (Implementació) En termes d'implementació, l'existència d'un *enllaç local* a $m()$ implica l'existència d'un objecte local en la implementació de l'operació $m()$.¹⁴

Exemple 16 (Enllaç dirigit local)

```
class A
{
    void m1()
    {
        B b = new B();          // 1
        b.f();
    }

    void m2()
    {
        B b = new B();          // 2
        b.f();
    }
}
```

La línia (1) diu que tot objecte $a:A$ té un enllaç dirigit $a:A \rightarrow b:B$ tal que el seu àmbit de localitat és (una realització de) l'operació $m1()$. La línia (2) diu que tot objecte $a:A$ té un enllaç dirigit $a:A \rightarrow b:B$ tal que el seu àmbit de localitat és (una realització de) l'operació $m2()$. Per tant no hi ha cap mena de relació (a part del nom) entre la b de la línia (1) i la b de la línia (2).

Donat un objecte $a:A$ el missatge $f()$ té com a contexts d'emissió possibles qualsevol realització de l'operació $m1()$ i qualsevol realització de l'operació $m2()$. Cadascun dels contextos d'emissió possibles és un context de localitat diferent.

¹²Pròpiament, donat un enllaç dirigit local tal que el seu context de localitat és una realització d'una operació $m()$, un cop aquesta realització de $m()$ acaba la seva tasca l'enllaç dirigit local deixa de ser vàlid.

¹³Pròpiament, l'enllaç local queda ocult dins de la reaalització pertinent de l'operació $m()$.

¹⁴ La implementació de les operacions en OO són mètodes; en programació estructurada són accions i funcions.

A diferència del que passava a l'exemple 15, pàgina 45, les crides $a.m1()$ i $a.m2()$ no comparteixen el mateix enllaç dirigit; en cada cas l'objecte destinació de l'enllaç és un objecte diferent. Això és així perquè cada crida és un context de localitat diferent.

El codi següent mostra dos contextos d'emissió diferents pel missatge $f()$ però que són realitzacions diferents d'una mateixa operació. En aquest cas tampoc hi ha compartició de l'enllaç dirigit emprat: cada context d'emissió té el seu propi objecte destinació de l'enllaç dirigit.

```
A a = new A();  
a.m1();           // 1  
a.m1();           // 2
```

2.3 Enllaç dirigit de paràmetre

Definició. Enllaç dirigit de paràmetre

Direm que un enllaç dirigit és de paràmetre ($a.m()$) quan:

- És un enllaç dirigit local ($a.m()$)
- L'objecte destinació de l'enllaç dirigit es propaga a través de la interfície del context de localitat $m()$

Publicació de la destinació. El llenguatge algorísmic que estem emprant ofereix dos mecanismes per fer visible informació a la interfície d'una operació: el valor de retorn de l'operació, o els arguments d'aquesta. Per tant, donat un enllaç de paràmetre ($a.m()$), l'objecte que constitueix la destinació de l'enllaç és o bé un dels arguments de l'operació $m()$, o bé és el seu valor de retorn.

Excursió. (Implementació) En termes d'implementació, l'existència d'un enllaç de paràmetre ($a.m()$) implica l'existència d'un paràmetre o d'un valor de retorn en la implementació de l'operació $m()$.¹⁵

Exemple 17 (Enllaç dirigit de paràmetre)

```
class A  
{  
    void m1(B b)           // 1  
    {  
        b.f();  
    }  
  
    void m2(B b)           // 2  
    {  
        b.f();  
    }  
}
```

La línia (1) diu que tot objecte $a:A$ té un enllaç dirigit $a:A \rightarrow b:B$ tal

¹⁵ La implementació de les operacions en OO són mètodes; en programació estructurada són accions i funcions.

3.2 Àmbit de validesa dels enllaços dirigits

J.M. Merenciano

que el seu àmbit de localitat és una realització de l'operació $m1()$. A més, la destinació de l'enllaç dirigit es publica a través de la interfície de $m1()$. La línia (2) diu que tot objecte $a:A$ té un enllaç dirigit $a:A \rightarrow b:B$ tal que el seu àmbit de localitat és una realització de l'operació $m2()$. A més, la destinació de l'enllaç dirigit es publica a través de la interfície de $m2()$. Per tant no hi ha cap mena de relació (a part del nom) entre la b de la línia (1) i la b de la línia (2).

El comportament, pel que fa a la validesa dels enllaços dirigits, és idèntic al de l'exemple 16, pàgina 47, \rightarrow on els enllaços dirigits són locals enlloc de ser de paràmetre.

El que canvia són les crides:

```
B b1 = new B();
B b2 = new B();

A a = new A();

a.m1(b1);           // 1
a.m1(b2);           // 2
a.m1(b1);           // 3
```

En el codi anterior, cada crida és un context de localitat diferent. Per tant en tots tres casos els enllaços dirigits emprats són diferents.

La comparació entre les crides (1) i (3) mereix especial atenció: tot i tenir contextos de localitat diferents, i per tant diferents enllaços dirigits, l'origen i la destinació dels enllaços coincideixen! El component (o la classe, si pensem en termes d'implementació) A s'ha definit de manera que en cada recepció del missatge $m1()$ es generi un nou enllaç dirigit. Ara bé, l'emissor del missatge $m1()$ imposa a través de la crida quin és la destinació d'aquest nou enllaç dirigit.¹⁶ \rightarrow Per això podem tenir enllaços dirigits conceptualment diferents, però idèntics a la pràctica.

Exemple 16. Enllaç dirigit local, pàgina 47

Exemple 18. Enllaç dirigit de paràmetre via argument, pàgina 49

Semàntica de la publicació via argument. Quan l'objecte destinació de l'enllaç és un argument del context de localitat $m()$,¹⁷ significa que l'emissor del missatge $m()$ imposa la destinació de l'enllaç dirigit que s'ha de crear localment a cada realització de $m()$.

Exemple 18 (Enllaç dirigit de paràmetre via argument) Siguin dos objectes $a:A$ i $b:B$. Suposem que en el codi apareix la crida $a.m(b)$. És a dir, l'objecte $a:A$ rep un missatge $m()$ amb argument l'objecte $b:B$.

Necessàriament el component A ha d'oferir l'operació $m()$, i aquesta ha d'esperar un argument que sigui realització del component B . Suposem que en la realització de $m()$ s'envia un missatge $m1()$ a l'argument rebut. Tot plegat porta a escriure el següent codi:

¹⁶Vegeu el paràgraf següent i l'exemple 18, a continuació.

¹⁷Hem abusat del llenguatge. Pròpiament el context de localitat és una realització de l'operació $m()$.

```
class A
{
    void m(B x)                // Sintaxi Java
    {
        ...
        x.f();
        ...
    }
}
```

El mecanisme del pas de paràmetres és qui vincula el paràmetre formal x amb el paràmetre real (b en l'exemple que estem considerant). Així, el codi de la classe A només diu que cada realització de $m()$ és un context de localitat per un enllaç dirigir $a : A \rightarrow x : B$ però no diu quina és la destinació d'aquest enllaç. És més, demana que sigui l'emissor del missatge $m()$ qui indiqui quina és la destinació de l'enllaç dirigit.

Semàntica de la publicació via retorn. Quan l'objecte destinació de l'enllaç és un valor de retorn del context de localitat $m()$,¹⁸ significa que l'emissor del missatge $m()$ rep, un cop acabada la tasca desencadenada per la recepció del missatge, quin és l'enllaç dirigit local emprat per aquesta realització concreta de l'operació $m()$.

Exemple 19 (Enllaç dirigit de paràmetre via valor de retorn)

Siguin dos objectes $a : A$ i $b : B$. Suposem que en el codi apareix la crida $b := a.m()$. És a dir, l'objecte $a : A$ rep un missatge $m()$ que retorna un objecte $b : B$.

Necessàriament el component A ha d'oferir l'operació $m()$, i aquesta ha de retornar un objecte que sigui realització del component B . Suposem que en la realització de $m()$ s'envia un missatge $m1()$ a l'objecte que finalment es retorna. Tot plegat porta a escriure el següent codi:

```
class A
{
    B m()                // Sintaxi Java
    {
        ...
        x.f();
        ...
        return x;
    }
}
```

Desconeixem quin és l'objecte x que apareix dins del codi de $m()$. (Bé, podríem llegir tot el codi i descobrir-ho; però i si només tenim la interfície de $m()$?!). És més, no ens interessa per a res. En termes de components, A ofereix una operació $m()$; el principi de l'Ocultació total deixa llibertat a

¹⁸Hem abusat del llenguatge. Pròpiament el context de localitat és una realització de l'operació $m()$.

3.2 Àmbit de validesa dels enllaços dirigits

J.M. Merenciano

*com defineix i gestiona A l'objecte que ha de retornar. Tot client de A en té prou en assumir que la crida a $m()$ retrona un objecte $b:B$. Al seu torn A , com a servidor, no li interessa en absolut què en farà el client de l'objecte $b:B$ que li proporciona. Per aquests motius és convenient considerar que $m()$ és l'àmbit de localitat de x . Això dóna la màxima llibertat tant a A com als seus clients.*¹⁹

2.4 La metàfora de la propietat oculta

Per facilitar la comprensió dels diferents àmbits de validesa dels enllaços dirigits usarem la metàfora de la *propietat* i l'assumpció que la validesa d'un objecte posseït depèn dels capricis del seu propietari. A més direm que una propietat és *oculta* quan només el propietari sap quines són les seves possessions. Llavors podem dir:

- **Enllaç dirigit d'atribut.** L'enllaç és "propietat oculta" de l'objecte emissor
- **Enllaç dirigit local.** L'enllaç és "propietat oculta" del context d'emissió, és a dir, de la realització concreta de l'operació que genera el missatge
- **Enllaç dirigit de paràmetre.** L'enllaç és "propietat oculta" del context d'emissió, però a més s'ofereixen els mecanismes per publicitar la propietat²⁰

(10)

Excursió. (Mecanisme de pas de paràmetres) La sintaxi que estem usant en les crides està inspirada en la de llenguatges com el Java, el C++ o el Pascal, però la semàntica no hi coincideix al cent per cent.

Independentment del mecanisme de pas de paràmetre que usi el llenguatge de la implementació, *en el disseny passem objectes*. No hem de pensar en còpies, ni apuntadors, ni referències. Hem de pensar en objectes, que poden ser vistos des de diferents llocs, i que poden rebre noms diferents segons des d'on són vistos.

Els codis que segueixen són equivalents en el disseny. En la impleentació l'equivalència d'aquests codis dependrà del llenguatge de programació. Per exemple, en Java, si `Elem` hereda d'`Object`, gràcies a l'ús de les referències els codis seran equivalents. En C++, si enlloc de manipular objectes manipulem apuntadors sobre aquests, si `Elem` és una `class`, i tots els constructors de còpia i els operadors d'assignació s'han defninit de tal manera

¹⁹ Alguns llenguatges de programació com el C++ o el Java permeten retornar qualsevol cosa. Altres llenguatges, com l'Eiffel o el Pascal, defineixen implícitament una variable local, de nom prefixat, que és l'única que es pot retornar. En aquests casos la localitat de l'enllaç dirigit de paràmetre és conseqüència de la semàntica del propi llenguatge, i no pas a criteris de desenvolupament o de disseny.

²⁰ És a dir, mostrem a tothom els títols de propietat.

que facin una còpia binària superficial (és a dir, no recursiva), els dos codis (convenientment adaptats) es comportaran igual. Si en C++ usem, com és habitual, una gestió de memòria on apareguin simultàniament apuntadors i referències, així com constructors per còpia o operadors d'assignació que realitzen còpies profundes (és a dir, recursives), llavors en la majoria dels casos (però no sempre) els dos codis es comportaran de manera diferent.

```
// ***** Codi 1 *****

a: Pila;                // Definim una pila
b: Elem;

a.empila(b);            // Posem un element al
                        // cim de la pila
b.modifica();           // Modifiquem l'element
                        // JA EMPILAT
b:= a.desempila();       // Treiem l'element de
                        // la pila

// ***** Codi 2 *****

a: Pila;                // Definim una pila
b: Elem;

b.modifica();           // Modifiquem l'element
a.empila(b);            // Posem un element
                        // MODIFICAT
                        // al cim de la pila
b:= a.desempila();       // Treiem l'element de
                        // la pila
```

3 Visibilitat

3.1 Visibilitat entre objectes

Definició. Visibilitat (directa) d'un objecte

Direm que un objecte oe té visibilitat directa sobre un objecte or si existeix un enllaç dirigit $oe \rightarrow or$.

En aquest cas direm que l'objecte or és visible des de l'objecte oe .



Notació. En el que segueix, i mentre no es digui el contrari, quan es parli de *visibilitat* el que de fet es vol dir és *visibilitat directa*.^{21,→}

²¹A l'apartat 3.3.4. Visibilitat indirecta, pàgina 62, s'introdueix la *visibilitat indirecta*.

Definició. Visibilitat d'atribut

Si existeix un enllaç dirigit d'atribut $oe \rightarrow or$ direm que oe té visibilitat d'atribut sobre or .

Definició. Visibilitat de paràmetre

Si existeix un enllaç dirigit de paràmetre $oe \rightarrow or$ direm que oe té visibilitat de paràmetre sobre or .

Definició. Visibilitat local

Si existeix un enllaç dirigit local $oe \rightarrow or$ (i que no és de paràmetre) direm que oe té visibilitat local sobre or .

Visibilitat, enllaços i àmbits. La visibilitat és l'existència d'un enllaç dirigit. Els diferents tipus de visibilitat (d'atribut, local o de paràmetre) es corresponen a l'àmbit de validesa dels enllaços dirigits.

Visibilitat estable. Si oe té visibilitat d'atribut sobre or vol dir que sempre que oe estigui realitzant alguna de les seves operacions té un enllaç dirigit vàlid per accedir a or . I de fet aquest enllaç dirigit sempre és el mateix. Així la visibilitat d'atribut és una visibilitat estable.

Visibilitat variable. Si oe té visibilitat local o de paràmetre sobre or vol dir que que la validesa de l'enllaç dirigit dependrà del context d'emissió del missatge, i de la realització concreta d'aquest context d'emissió. És a dir, la visibilitat local i la visibilitat de paràmetre són visibilitats variables: l'existència o no de l'enllaç dirigit (o de la seva validesa) depèn de quina operació s'està realitzant en el moment de voler usar l'enllaç dirigit (i més concretament, de la realització concreta d'aquesta operació).

Visibilitat i existència. La visibilitat i l'existència són dos conceptes diferents. Un objecte pot existir, però no ser visible en un determinat context perquè en aquest context no hi ha cap enllaç dirigit vàlid que el tingui com a destinació de l'enllaç.

Exemple 20 (Objectes invisibles) En l'exemple 12, pàgina 41, es parla dels objectes zombies. Es tracta d'objectes existents però no visibles per ningú.

Exemple 21 (Visibilitat i existència) Durant la realització d'una operació $m()$, l'objecte oe que ha rebut el missatge $m()$ té visibilitat sobre els següents objectes:

- Objectes dels quals oe en té visibilitat d'atribut
- Objectes dels quals oe en té visibilitat de paràmetre a $m()$

- Objectes dels quals oe en té visibilitat local a $m()$

Però en canvi, per exemple, no té visibilitat (a no ser que l'hagi rebut per paràmetre) de l'objecte emissor del missatge $m()$.

Excursió. (Altres tipus de visibilitat) Els diferents llenguatges de programació ofereixen d'altres tipus de visibilitat, que no contemplem aquí. Dos dels més habituals són els següents:

- **Global** Direm que sobre un objecte or existeix *visibilitat global* si per qualsevol objecte oe existeix un *enllaç dirigit* de oe a or .

Cal observar com la *visibilitat global* assegura l'existència de l'enllaç dirigit $oe \rightarrow or$, tot i que aquest necessàriament serà diferent per cada oe .

El principi de l'*Encapsulament* i l'*Ocultació*, que en el disseny els considerem regles de funcionament, impedeixen la possibilitat de la *visibilitat global*.

- **Component** Direm que sobre un objecte or existeix *visibilitat de component* si tots els objectes oe d'un mateix component A comparteixen un mateix *enllaç dirigit* $oe : A \rightarrow or$. S'acostuma a expressar com $A \rightarrow or$.

La *visibilitat de component* assegura que tots els objectes que siguin realització d'un mateix component A tenen accés a un mateix objecte or ; i que el mecanisme d'accés és el mateix.

3.2 Visibilitat entre components

3.2.1 La visibilitat com a potencialitat

Potencialitat de col·laboració. La presència de la *visibilitat* és el que permet la col·laboració entre els objectes: si oe té visibilitat sobre or significa que existeix un enllaç dirigit $oe \rightarrow or$ que permet que oe envii, dins l'àmbit de validesa de l'enllaç $oe \rightarrow or$, un missatge a or .

Encapsulació de la visibilitat en el component. Aquesta potencialitat de col·laboració que expressa la visibilitat ha d'estar encapsulada en el component del qual oe n'és realització. És a dir, per poder enviar un missatge a or , l'objecte oe n'ha de tenir visibilitat; i per tant el component corresponent ha d'oferir la possibilitat d'aquesta visibilitat. D'aquí la definició de *visibilitat entre components*.

Definició. Visibilitat entre components

Diem que un component A té visibilitat d'un component B si i només si alguna realització de A té visibilitat d'alguna realització de B .

Notació. Expressarem la visibilitat d'un component A sobre un component B amb $A \rightarrow B$



Visibilitat entre components en termes de la visibilitat d'objectes. El que diu la definició és que si un component A té visibilitat d'un component B és perquè existeixen dos objectes $oe:A$ i $or:B$ tals que existeix un enllaç dirigit $oe:A \rightarrow or:B$; és a dir, existeixen dos objectes $oe:A$ i $or:B$ tals que $oe:A$ té visibilitat sobre $or:B$. Formalment:

$$A \rightarrow B = (\exists oe:A)(\exists or:B)(oe:A \rightarrow or:B)$$

No obligatorietat de la realització de la visibilitat. La definició de la visibilitat $A \rightarrow B$ diu que les realitzacions de A tenen la potencialitat de mantenir enllaços dirigits cap a realitzacions de B , però no exigeix la realització d'aquesta potencialitat. Aquesta no exigència és el que diu el primer quantificador: existencial i no pas universal.

Obligatorietat de la realització de la visibilitat. La definició de la visibilitat $A \rightarrow B$ no exigeix a un objecte $a:A$ concret que la realitzi, però sí que exigeix que alguna realització de A ho faci.

Enllaç dirigit i visibilitat de component: dues cares d'una mateixa moneda. La obligatorietat que algú realitzi la visibilitat concorda amb la visió dels components i els objectes com a dues cares una mateixa moneda: un no pot existir sense l'altre, i a la inversa. \rightarrow Ara el que tenim és que enllaç dirigit i visibilitat (de component) són les dues cares d'una mateixa moneda: l'enllaç dirigit existeix perquè la visibilitat li ho permet; la visibilitat existeix perquè es realitza en algun enllaç.

3.1.2.1. Assignació de responsabilitats, pàgina 37

La visibilitat $A \rightarrow B$ diu que:

- *Realització concreta no obligada:* Cada objecte $a:A$ pot tenir visibilitat sobre algun objecte $b:B$
- *Realització del conjunt obligada:* Com a mínim existeix un objecte $a:A$ que té visibilitat sobre algun objecte $b:B$

(11)

Excursió. (Visibilitat obligada i visibilitat optativa) Sovint es consideren dos tipus de visibilitat: la *obligada* i la *no obligada* o *optativa*.

La visibilitat *no obligada* és la presentada aquí: permet que donada una visibilitat $A \rightarrow B$ algunes realitzacions de A no la realitzin; és a dir, que per algunes realitzacions de A no hi hagi cap realització de B sobre les que aquestes realitzacions de A en tinguin visibilitat. Formalment:

$A \rightarrow B$ admet $(\exists oe:A)(\exists or:B) \neg (oe:A \rightarrow or:B)$
exigeix $(\exists oe:A)(\exists or:B)(oe:A \rightarrow or:B)$

La *visibilitat obligada* converteix la potencialitat en obligatorietat: donada una visibilitat obligada $A \rightarrow B$ tota realització de A ha de tenir visibilitat sobre alguna realització de B. Formalment:

$$A \rightarrow B = (\forall oe:A)(\exists or:B)(oe:A \rightarrow or:B)$$

La *visibilitat obligada* es pot veure com un cas particular de la *visibilitat* (no obligada) que hem presentat, però no a la inversa.²², \rightarrow

$$(\forall oe:A)(\exists or:B)(oe:A \rightarrow or:B) \Rightarrow (\exists oe:A)(\exists or:B)(oe:A \rightarrow or:B)$$

3.2.2 La visibilitat com a abstracció

Enllaços dirigits com a realització de la visibilitat. L'existència d'un enllaç dirigit és la realització d'una potencialitat; en concret de la potencialitat que expressem amb la visibilitat. Per tant no hi pot haver enllaç dirigit sense la visibilitat que la permeti. (Així podem escriure $oe:A \rightarrow or:B \Rightarrow A \rightarrow B$).

Visibilitat com a abstracció dels enllaços dirigits. Les definicions, i la notació, tal com s'han presentat, permeten veure la *visibilitat* $A \rightarrow B$ com una *abstracció* que expressa les potencialitats comunicatives; la *realització* d'aquesta abstracció són els *enllaços dirigits* $oe:A \rightarrow or:B$.

La visibilitat i l'abstracció dels enllaços dirigits. La visió de la visibilitat com a *abstracció* dels enllaços dirigits pot portar a intentar veure-la com el conjunt de tots els enllaços dirigits presents o possibles. Aquesta visió, tot i no ser errònia, oblida un element fonamental de la visibilitat $A \rightarrow B$: tot objecte realització de A la pot realitzar, però com a mínim un l'ha de realitzar. \rightarrow

3.2.3 Àmbits de visibilitat

Definició. Visibilitat d'atribut

Si existeix un enllaç dirigit d'atribut $oe:A \rightarrow or:B$ direm que A té visibilitat d'atribut sobre B.

Definició. Visibilitat de paràmetre

Si existeix un enllaç dirigit de paràmetre $oe:A \rightarrow or:B$ direm que A té visibilitat de paràmetre sobre B.

²²Per ser correcta la implicació cal que el domini del quantificador universal no sigui buit; la qual cosa en aquest cas significa que com a mínim existeixi un objecte que sigui realització del component A. Aquesta existència queda assegurada per la conclusió \rightarrow , que diu que els components sense objectes no tenen sentit.

Definició. Visibilitat local

Si existeix un enllaç dirigit local $oe:A \rightarrow or:B$ (i que no és de paràmetre) direm que A té visibilitat local sobre B .

Visibilitat, enllaços i àmbits. La visibilitat (entre objectes o components) és l'existència d'un enllaç dirigit. Els diferents tipus de visibilitat (d'atribut, local o de paràmetre) es corresponen a l'àmbit de validesa dels enllaços dirigits.

Potencialitats de la capacitat d'atribut. La *visibilitat d'atribut* de A a B diu que el component A ofereix als seus objectes la capacitat de mantenir, cadascun d'ells, un enllaç dirigit cap a un objecte realització de B , que manté la seva validesa al llarg de tota la vida de l'objecte origen de l'enllaç dirigit.

Potencialitats de la capacitat local. La *visibilitat local* de A a B diu que el component A ofereix als seus objectes la capacitat de mantenir, cadascun d'ells i per cadscuna de les realitzacions de les operacions ofertes pel component A , un enllaç dirigit cap a un objecte realització de B .

Potencialitats de la capacitat de paràmetre. La *visibilitat de paràmetre* de A a B diu que el component A ofereix als seus objectes la capacitat de mantenir, cadascun d'ells i per cadscuna de les realitzacions de les operacions ofertes pel component A , un enllaç dirigit cap a un objecte realització de B (a l'igual que la visibilitat local), així com la capacitat de comunicar a la interfície de les operacions la destinació de l'enllaç dirigit.

3.3 Multiplicitat de la visibilitat

3.3.1 Multiplicitat dels enllaços dirigits

Múltiples destinacions. Un cas especial dels enllaços dirigits és aquell en el es permet accedir a diferents objectes destinació a partir d'un mateix objecte origen de l'enllaç. Són els anomenats *enllaços dirigits multiavaluats*.

Definició. Enllaç dirigit multiavaluat

Anomenem enllaç dirigit multiavaluat aquell enllaç dirigit que té més d'una destinació.

Notació. Escriurem $a \rightarrow [x, y, z, \dots]$ per expressar un enllaç dirigit multiavaluat de l'objecte a als objectes x, y, z, \dots



Definició. Enllaç dirigit monoavaluat o simple

Tot enllaç dirigit que no sigui multiavaluat direm que és un enllaç dirigit monoavaluat o simple.

Exemple 22 (Enllaç dirigit multiavaluat)

Tota etapa d'una caminada té dos indrets singulars: l'indret d'inici i l'indret de finalització de l'etapa. Suposem que el disseny requereix que l'objecte $e:Etapa$ ha d'enviar el missatge `visitat()` als seus indrets singulars cada cop que algun participant en la caminada passa pel control de final d'etapa.

Una manera de resoldre-ho seria assegurant per cada $e:Etapa$ un enllaç dirigit $e \rightarrow [or, fi]$, on or i fi són els objectes que representen els indrets d'inici i final de l'etapa e . Llavors el missatge `visitat()` sobre aquest enllaç dirigit arribarà als dos indrets singulars de l'etapa: l'inici i la destinació.

Exemple 23 (Semàntica d'un enllaç dirigit multiavaluat) *En l'exemple 22 \rightarrow cada etapa s'enllaça amb els seus indrets singulars; tenim un enllaç dirigit biavaluat amb la semàntica²³, \rightarrow "indrets singulars de l'etapa".*

Val la pena comparar aquesta semàntica amb les semàntiques donades a l'exemple 13, pàgina 42 \rightarrow .

Exemple 22. Enllaç dirigit multiavaluat, pàgina 58

3.1.4. Enllaços dirigits, pàgina 40

Exemple 13. Semàntica d'un enllaç dirigit, pàgina 42

Restricció dels enllaços dirigits multiavaluats. Per simplificar els raonaments i tot el disseny, només admetrem enllaços dirigits multiavaluats homogenis; és a dir, aquells tals que totes les destinacions de l'enllaç dirigit són realitzacions d'un mateix component.²⁴



Notació. Usarem $a \rightarrow B^*$ per expressar un enllaç dirigit multiavaluat de l'objecte a a un conjunt de realitzacions de B

Enllaç dirigit multiavaluat versus múltiples enllaços simples. L'agregació de diferents enllaços dirigits simples, amb un mateix objecte origen, en un enllaç dirigit multiavaluat només és útil en el cas que els diferents enllaços dirigits tinguin una mateixa semàntica. Altrament és preferible parlar de múltiples enllaços dirigits simples.

Exemple 24 (Agregació o no de d'enllaços dirigits)

*Suposem un objecte $g:GràficaMatemàtica$ i una operació de nom `dibuixarGràfica()` que localment crea diferents punts d'interès: màxims i mínims locals, punts d'inflexió, asímptotes, etc., per tal de poder-los representar en pantalla. Llavors podem dir que l'objecte $g:GràficaMatemàtica$ té un enllaç local multiavaluat sobre **Punt**, amb la semàntica `puntsInterès`.*

²³La semàntica dels enllaços dirigits s'ha introduït a l'apartat 12., pàgina 42, de 3.1.4. Enllaços dirigits, pàgina 40.

²⁴Aquesta restricció, entre d'altres coses, permet simplificar la semàntica dels enllaços dirigits.

*Ara bé, segons com sigui el tractament intern d'aquests punts d'interès seria preferible parlar de diferents enllaços dirigits multiavaluats, tots sobre **Punt** ped amb diferents semàntiques: *MaxMin*, *Inflexions*, i *Asímtotes*.*

El criteri que ens fa parlar d'un sol enllaç dirigit (multiavaluat) o de diversos enllaços dirigits (multiavaluats o simples) és semàntic:²⁵ en funció del significat o la funcionalitat que donem a cada visibilitat les agruparem o no.

Missatge sobre un enllaç dirigit multiavaluat. En el disseny tot enllaç dirigit és un canal a través del qual enviar missatges. En el cas que l'enllaç dirigit sigui multiavaluat entenem que el missatge arriba a tots els objectes que són destinació de l'enllaç dirigit. És un sol missatge amb diferents destinataris.

Excursió. (Implementació) La²⁶, \rightarrow majoria de llenguatges de programació no admeten els enllaços dirigits multiavaluats, la qual cosa obliga a reconvertir cada enllaç dirigit multiavaluat en múltiples enllaços dirigits monoavaluats. Així un missatge sobre un enllaç dirigit multiavaluat caldrà implementar-lo com múltiples còpies d'un mateix missatge, cadascuna d'elles enviada a través d'un enllaç dirigit diferent.

5.4.Incursió en la implementació, pàgina 97

Els llenguatges que admeten enllaços dirigits multiavaluats no acostumen a admetre'ls com a canal pels missatges. En aquest cas cal enviar individualment una rèplica del missatge a cadascuna de les destinacions de l'enllaç.

Tant en un cas com en un altre, l'ús en el disseny d'enllaços dirigits multiavaluats comporta, en implementació, la replicació dels missatges. En alguns entorns aquesta replicació pot ser transparent a l'usuari.

Malgrat aquesta discrepància de tractament, l'ús dels enllaços dirigits multiavaluats en el disseny és beneficiosa per la compacitat i simplicitat que dona.

Generalització del concepte d'enllaç dirigit. Els *enllaços dirigits simples* es poden veure com un cas particular dels *enllaços dirigits multiavaluats*. Aquesta observació permet la següent definició:

Definició. Enllaç dirigit (generalitzat)

Diem que hi ha un enllaç dirigit entre l'objecte $a : A$ (l'origen de l'enllaç) i un conjunt (anomenat destinació de l'enllaç dirigit) de realitzacions del component B quan:

- *a té coneixement de l'existència de la destinació de l'enllaç dirigit*
- *a té un mecanisme per accedir als objectes que constitueixen la destinació de l'enllaç dirigit (és a dir, un mecanisme per fer-lis peticions, per enviar-lis missatges).*

²⁵I sovint força subjectiu.

²⁶A l'apartat 5.4.Incursió en la implementació, pàgina 97 es tracten els aspectes d'implementació amb més profunditat i rigor.

Si la destinació de l'enllaç dirigit conté un sol objecte direm que l'enllaç dirigit és monoavaluat. Altrament direm que l'enllaç dirigit és multiavaluat.

! (6) 🔍

Notació. A partir d'ara, i mentre no es digui el contrari, quan es parli d'enllaç dirigit cal entendre un enllaç dirigit generalitzat; és a dir, que tant pot ser *monoavaluat* com *multiavaluat*.

Així, $a \rightarrow b$ cal veure'l com un abreujament notacional de $a \rightarrow [b]$.

(12)

En el disseny, tot missatge arriba a tots els destinataris de l'enllaç dirigit (generalitzat) emprat.

3.3.2 Multiplicitat de la visibilitat d'objectes

Definició. Visibilitat multiavaluada (d'un objecte)

Direm que un objecte a té una visibilitat multiavaluada sobre un component B si existeix un enllaç dirigit multiavaluat de a a objectes que són realització de B .



Notació. Escriurem $a \rightarrow B^*$ per expressar una visibilitat multiavaluada de l'objecte a sobre el component B .

Excursió. (Visibilitat multiavaluada d'objecte) La definició de la visibilitat multiavaluada d'un objecte la podem expressar com:

$a \rightarrow B^* \iff \exists x:B, y:B, z:B, \dots$ tal que existeix l'enllaç dirigit $a \rightarrow [x:B, y:B, z:B, \dots]$



Notació. Si en l'expressió d'una *visibilitat multiavaluada* enlloc d'usar l'exponent $*$ usem un exponent numèric, aquest indica el nombre màxim d'objectes que poden ser destinació dels enllaços dirigits que conformen la visibilitat en qüestió. Per exemple $a \rightarrow A^2$ expressa una visibilitat biavaluada: els seus enllaços dirigits tenen com a molt 2 objectes destinació de l'enllaç.

Exemple 25 (Visibilitat multiavaluada d'objecte) En l'exemple 22, pàgina 58, cada objecte $e:Etapa$ té una visibilitat biavaluada (multi = 2) sobre $Indret$. Ho expressarem amb $e \rightarrow Indret^2$.

Exemple 26 (Agregació o no de visibilitats (d'objecte))

En l'exemple 24, el primer disseny significa que l'objecte $g: \text{GràficaMatemàtica}$ té visibilitat local multiavaluada sobre Punt ($\text{puntsInterès}: g \rightarrow \text{Punt}^*$). En el segon disseny el que es té són diferents visibilitats multiavaluades: $\text{maxMin}: g \rightarrow \text{Punt}^*$, $\text{inflexions}: g \rightarrow \text{Punt}^*$, i $\text{asíptotes}: g \rightarrow \text{Punt}^*$.

Visibilitat generalitzada (d'un objecte). L'ús dels enllaços dirigits generalitzats permet fusionar els conceptes de visibilitat entre objectes i visibilitat multiavaluada d'un objecte

Definició. Visibilitat generalitzada (d'objecte)

Direm que un objecte a té visibilitat (directa, generalitzada) sobre un component B si existeix un enllaç dirigit generalitzat de $a \rightarrow B^*$.

Notació. En tot el que segueix, i mentre no es digui el contrari, la visibilitat (d'objecte) sempre serà una visibilitat generalitzada



Excursió. (Sobre la notació) La notació introduïda usa l'exponenciació per indicar el nombre d'objectes que són destinacions d'un mateix enllaç dirigit: un valor numèric per indicar un màxim concret, i l'asteric (*) per indicar un valor indefinit. D'aquesta manera es potencia la visió de la visibilitat monoavaluada com un cas particular de la visibilitat multiavaluada: $a \rightarrow B^1 = a \rightarrow B$ (l'exponent unitari no s'escriu).

3.3.3 Multiplicitat de la visibilitat de components

Definició. Visibilitat multiavaluada (d'un component)

Direm que un component A té una visibilitat multiavaluada sobre un component B si existeix un objecte $a: A$ amb visibilitat multiavaluada sobre B .

Notació. Escriurem $A \rightarrow B^*$ per expressar una visibilitat multiavaluada del component A sobre el component B .



Excursió. (Visibilitat multiavaluada d'un component) La definició de la visibilitat multiavaluada d'un component la podem expressar com:

$A \rightarrow B^* == \exists a: A, \text{ tal que existeix l'enllaç dirigit multiavaluat d'objecte } a \rightarrow B^*$

Exemple 27 En l'exemple 25, pàgina 60 (o el que és el mateix, en l'exemple 22, pàgina 58), el que tenim és una visibilitat $Etapa \rightarrow Indret^2$.

Exemple 28 (Agregació o no de visibilitats (de component))

L'exemple 26 planteja l'anàlisi de les visibilitats d'objecte de l'exemple 24. En termes de les visibilitats de component, en el primer disseny el component *GràficaMatemàtica* té una visibilitat local multiavaluada sobre *Punt* ($puntsInterès: GràficaMatemàtica \rightarrow Punt^*$); en el segon disseny el que es té són diferents visibilitats multiavaluades:

- $maxMin: GràficaMatemàtica \rightarrow Punt^*$
- $inflexions: GràficaMatemàtica \rightarrow Punt^*$
- $asímtotes: GràficaMatemàtica \rightarrow Punt^*$

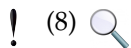
Visibilitat generalitzada (d'un component). L'ús dels enllaços dirigits generalitzats permet fusionar els conceptes de *visibilitat entre components* i *visibilitat multiavaluada d'un component*.

Definició. Visibilitat generalitzada (d'un component)

Diem que un component *A* té visibilitat (directa, generalitzada) sobre un component *B* si i només si existeix una realització de *A* amb visibilitat (directa, generalitzada) sobre el component *B*.



Notació. Si en l'expressió d'una *visibilitat multiavaluada* enlloc d'usar l'exponent $*$ usem l'exponent $+$ estem indicant que la visibilitat és obligada. Per exemple $A \rightarrow B^+$ expressa que per tot objecte $a:A$ hi ha un enllaç dirigit $a \rightarrow B^*$.



Notació. En el que segueix, i mentre no es digui el contrari, la visibilitat (d'un component) sempre serà una visibilitat generalitzada.

3.4 Visibilitat indirecta

3.4.1 Indirecció i enllaços dirigits

Definició. Enllaços dirigits indirectes

Direm que existeix un enllaç dirigit indirecte²⁷ amb origen $a:A$ i destinació una o diverses realitzacions de *B* si:

- Enllaç directe: Existeix un enllaç dirigit $a:A \rightarrow B^*$, o
- Enllaç amb intermediaris: Existeix un objecte $c:C$, que anomenarem intermediari, tal que

- Existeix un enllaç dirigit monavaluat $a : A \rightarrow c : C$,
- $c : C$ és l'origen d'un enllaç dirigit indirecte que té com a destinació una o més realitzacions de B

Notació. Usarem $a : A \xrightarrow{*} b : B^*$ per expressar l'existència d'un enllaç dirigit indirecte amb origen $a : A$ i destinacions sobre realitzacions de B .



Generalització de l'enllaç dirigit. Els enllaços dirigits indirectes són una generalització dels enllaços dirigits (també anomenats *directes*): la primera condició de la definició diu que tot enllaç dirigit és un enllaç dirigit indirecte.

Definició recursiva. Els enllaços dirigits (directes) esdevenen el *cas base* de la recursió que defineix els enllaços dirigits indirectes. El cas recursiu postula l'existència d'un objecte *intermediari* que fa de pont entre l'origen i les destinacions de l'enllaç dirigit indirecte.

Definició iterativa. En aplanar la definició recursiva s'obté que l'existència d'un d'enllaç dirigit indirecte $a : A \xrightarrow{*} B^*$ significa l'existència de k intermediaris ($c_1 : C_1, \dots, c_k : C_k$) i $k + 1$ enllaços dirigits. En concret han d'existir els enllaços dirigits $a : A \rightarrow c_1 : C_1, c_1 : C_1 \rightarrow c_2 : C_2, \dots, c_{k-1} : C_{k-1} \rightarrow c_k : C_k, c_k : C_k \rightarrow B^*$.

Aspectes de la notació. La notació $a : A \xrightarrow{*} B^*$ usa un asterisc (*) sobre el símbol de l'enllaç dirigit (\rightarrow) per expressar la idea d'iteració sobre l'accés: per tal d'accedir des de l'origen de l'enllaç a la seva destinació (o destinacions) s'ha de passar per diferents intermediaris, i per tant s'itera l'accés (cal un accés per cada intermediari). També es pot veure com l'existència de diferents enllaços dirigits: l'enllaç dirigit indirecte és un conjunt d'enllaços dirigits amb les propietats que els dona la definició iterativa presentada.

3.4.2 Cadena d'enllaços dirigits

La cadena de la indirecció. La definició iterativa d'un enllaç dirigit indirecte mostra com l'existència d'un $a : A \xrightarrow{*} B^*$ significa l'existència d'una *cadena d'enllaços dirigits*:

Definició. Cadena d'enllaços dirigits

Una cadena d'enllaços dirigits és una seqüència ordenada d'enllaços dirigits on cada enllaç dirigit és un enllaç dirigit monoavaluat que té com a destinació l'origen del següent enllaç dirigit de la cadena; a excepció del darrer enllaç dirigit de la cadena, que pot ser multiavaluat.

²⁷Presentem les definicions en termes dels enllaços dirigits generalitzats; fàcilment es poden refer en termes dels enllaços dirigits simples i en termes dels enllaços dirigits multiavalats.



Notació. Per expressar la *cadena d'enllaços dirigits* construïda per la seqüència $a:A \rightarrow c_1:C_1 + c_1:C_1 \rightarrow c_2:C_2, \dots + c_{k-1}:C_{k-1} \rightarrow c_k:C_k + c_k:C_k \rightarrow b:B^*$ escriurem $a:A \rightarrow c_1:C_1 \rightarrow \dots \rightarrow c_k:C_k \rightarrow b:B^*$.

Exemple 29 (Enllaços dirigits indirectes) Suposem que $a:A$ té un enllaç dirigit d'atribut $b:B$; i que aquest té un enllaç dirigit multiavaluat de paràmetre sobre C . Ho podem expressar com: $a \rightarrow b \rightarrow C^*$; o més succintament $a \rightarrow b \rightarrow C^*$. En aquest cas direm que $a:A$ té un enllaç dirigit indirecte sobre C , i ho expressarem amb $a:A \rightarrow C^*$.

Cal observar com la definició de visibilitat indirecta exigeix que tots els enllaços intermedis siguin monoavaluats (en aquest cas l'enllaç $a \rightarrow b$); només el darrer pot ser multiavaluat²⁸ (en aquest cas l'enllaç $a \rightarrow C^*$).

Cadena d'enllaços dirigits i visibilitat. L'existència d'un enllaç dirigit indirecte $a:A \rightarrow C^*$ implica l'existència d'un conjunt d'enllaços dirigits que constitueixen una cadena; però aquests enllaços dirigits només poden existir com a realització de les visibilitats ofertes pels components. És a dir, l'existència d'un enllaç dirigit indirecte $a:A \rightarrow C^*$ significa l'existència d'un conjunt de visibilitats que constitueixen una seqüència:²⁹ $A \rightarrow C_1 + \dots + C_k \rightarrow B$. En aquesta seqüència totes les visibilitats són monoavaluades, excepte la darrera, que és generalitzada.

Exemple 30 (Cadena d'enllaços dirigits i de visibilitats) En l'exemple 29, la presència de l'enllaç dirigit indirecte $a:A \rightarrow C$ significa l'existència de les visibilitats $A \rightarrow B$ i $B \rightarrow C^*$.

Per similitud amb els enllaços dirigits indirectes podem estar temptats d'expressar aquesta seqüència en termes de cadena: $A \rightarrow B \rightarrow C^*$. Però això no sempre serà possible, com s'analitza a continuació.

Seqüència de visibilitats sense cadena d'enllaços dirigits. Donada una seqüència enllaçable³⁰ de visibilitats no necessàriament significa l'existència d'una cadena d'enllaços dirigits: cada visibilitat de la cadena és un permís per mantenir un enllaç dirigit; però l'encadenament dels permisos no significa l'encadenament de les realitzacions d'aquests permisos.

²⁸Definició alternativa: tots els enllaços intermedis són monoavaluats; el darrer enllaç és un enllaç generalitzat.

²⁹Parlem de seqüència i no pas de cadena perquè com veurem tot seguit per poder parlar de cadena de visibilitats exigirem una condició més sobre la seqüència.

³⁰Aquí ens movem sobre idees intuïtives, i deixem les definicions per més endavant. L'enllaçabilitat de la cadena indica que la "desstinació" d'una visibilitat coincideix amb l'"origen" de la següent visibilitat de la seqüència.

Exemple 31 (Seqüència de visibilitats trencada) Sigui la seqüència de visibilitats $A \rightarrow B + B \rightarrow C$.³¹ La primera visibilitat de la seqüència significa que com a mínim un objecte $a : A$ té un enllaç dirigit cap algun objecte realització de B . Podem per tant suposar l'existència d'un enllaç dirigit $a : A \rightarrow b : B$ per algun $a : A$ i algun $b : B$.

La segona visibilitat de la seqüència significa que com a mínim un objecte $x : B$ té un enllaç dirigit cap algun objecte realització de C . Però res no assegura que aquest objecte sigui justament l'objecte $b : B$!! L'únic que podem afirmar és l'existència d'un enllaç dirigit $x : B \rightarrow c : C$ per algun $x : B$ i algun $c : C$.

Sense més informació que la que ens ofereix l'existència de la seqüència de visibilitats l'únic que podem afirmar és l'existència dels enllaços dirigits $a : A \rightarrow b : B$ i $x : B \rightarrow c : C$, que no constitueixen una cadena d'enllaços dirigits: la seqüència de visibilitats es trenca (no constitueix una cadena) en les seves realitzacions en forma d'enllaços dirigits.

És possible l'existència d'uns objecte $u : A$, $v : B$, $z : C$ tals que existeix la cadena d'enllaços dirigits $u : A \rightarrow v : B \rightarrow z : C$, però no tenim prou informació per afirmar-ho.

Excursió. (Condicció suficient per a la continuïtat) L'exemple 31 mostra com les seqüències de visibilitats poden trencar-se a nivell de les seves realitzacions: ningú no assegura l'existència d'una cadena d'enllaços dirigits tal que cada enllaç dirigit de la cadena sigui la realització de la visibilitat que ocupa la mateixa posició en la seqüència.

Una condició suficient per assegurar l'existència de cadenes d'enllaços dirigits compatible amb una seqüència de visibilitats donada és exigir que totes les visibilitats de la seqüència siguin obligades.³²

3.4.3 Indirecció i cadenes de visibilitat

Definició. Visibilitat indirecta

Un component³³ A té visibilitat indirecta sobre un component B si per algun objecte $a : A$ existeix un enllaç dirigit indirecte $a : A \rightarrow^* B^*$.

Notació. Usarem $A \rightarrow^* B^*$ per expressar una visibilitat indirecta de A sobre B .



³¹Per simplificar l'exemple considerem que la darrera visibilitat és monoavaluada; el mateix obtindríem amb una visibilitat multiavaluada.

³²De fet la primera visibilitat de la seqüència no cal que sigui obligada.

³³Presentem la definició de visibilitat indirecta de component. Anàlogament es pot parlar de la visibilitat indirecta d'objecte.

Aspectes de la notació. La notació proposada usa l'asterisc (*) per inidcar iteració. En la *visibilitat multiavaluada* la notació $(A \rightarrow B^*)$ expressa una *iteració en la destinació* de la visibilitat; això es correspon a la idea que un mateix enllaç permet accedir a més d'un objecte destinació. En la *visibilitat indirecta* la notació $(A \rightarrow^* B)$ expressa una *iteració en la visibilitat*; això es correspon a la idea que cal iterar el mecanisme d'accés al llarg de la cadena de visibilitats per arribar a la destinació.

Cadena de visibilitats. La visibilitat indirecta significa l'existència d'una cadena de visibilitats directes monoavaluades: cada component té visibilitat directa monoavaluada d'un altre component; amb excepció del darrer element de la cadena on no s'exigeix que la visibilitat sigui monoavaluada.

Exemple 32 (Visibilitat indirecta) Suposem que $a : A$ té un enllaç dirigit d'atribut sobre $b : B$; i que aquest té un enllaç dirigit multiavaluat de paràmetre sobre C . Ho podem expressar com: $a \rightarrow b \rightarrow b \rightarrow C^*$; o més sucintament $a \rightarrow b \rightarrow C^*$. En aquest cas direm que A té visibilitat indirecta sobre C , i ho expressarem amb $A \rightarrow^* C$.

Cal observar com la definició de visibilitat indirecta exigeix que totes les visibilitats intermèdies siguin monoavaluades (en aquest cas la visibilitat $A \rightarrow B$); només la darrera visibilitat pot ser multiavaluat³⁴ (en aquest cas la visibilitat $A \rightarrow C^*$).

Cadena de visibilitats i cadena d'enllaços dirigits. L'existència d'una cadena de visibilitats $A \rightarrow X_1 \rightarrow \dots \rightarrow X_k \rightarrow B^*$ significa l'existència d'un conjunt d'objectes $a : A, x_1 : X_1, \dots, x_k : X_k, b : B$ que constitueixen una *cadena d'enllaços dirigits* $a : A \rightarrow x_1 : X_1 \rightarrow \dots \rightarrow x_k : X_k \rightarrow b : B^*$, on cada enllaç dirigit és la realització de la potencialitat expressada per la visibilitat corresponent de la cadena. Per exemple, l'enllaç dirigit $x_i : X_i \rightarrow x_{i+1} : X_{i+1}$ és una realització de la visibilitat $X_i \rightarrow X_{i+1}$.

Realització d'una visibilitat indirecta. La visibilitat indirecta $A \rightarrow^* B^*$ expressa la potencialitat que tot objecte $a : A$ pugui tenir enllaços dirigits indirectes $a : A \rightarrow^* B^*$, i que com a mínim hi hagi un objecte $x : A$ que realitzi aquesta potencialitat.

Col·laboració en una cadena de visibilitats. Si tenim la visibilitat $A \rightarrow^* B^*$ significa que hi ha una *cadena de visibilitats directes* $A \rightarrow X_1 \rightarrow \dots \rightarrow X_k \rightarrow B^*$; i per tant una *cadena d'enllaços dirigits* $a : A \rightarrow x_1 : X_1 \rightarrow \dots \rightarrow x_k : X_k \rightarrow b : B^*$ que la realitza. D'aquesta manera l'objecte $a : A$ pot enviar un missatge $m_1()$ a l'objecte $x_1 : X_1$ a través de l'enllaç dirigit que realitza la visibilitat $A \rightarrow X_1$. En rebre aquest missatge l'objecte x_1 pot enviar un missatge $m_2()$ a l'objecte $x_2 : X_2$ a través de l'enllaç dirigit que realitza la visibilitat $X_1 \rightarrow X_2$. Aquest procés es pot anar repetint fins que finalment l'objecte $x_k : X_k$ envia un missatge $m_{k+1}()$ a

³⁴Definició alternativa: tots els enllaços intermedis són monoavaluats; el darrer enllaç és un enllaç generalitzat.

través de l'enllaç dirigit que realitza la visibilitat $X_k \rightarrow B^*$. En definitiva, l'objecte $a : A$ ha emprat la *visibilitat indirecta* $A \xrightarrow{*} B^*$ (més concretament, la realització d'aquesta en termes de l'enllaç dirigit indirecte $a : A \xrightarrow{*} B^*$) per enviar un missatge $m_i()$ a un conjunt d'objectes que són realització del component B .

Col·laboració a través de la visibilitat indirecta. Com acabem de veure la visibilitat indirecta $A \xrightarrow{*} B^*$ és una cadena de visibilitats directes; i es pot usar aquesta cadena perquè els objectes que són realització del primer component de la cadena (el component A) puguin enviar missatges als objectes que són realització del darrer component de la cadena (B). Així la *visibilitat indirecta* es comporta exactament igual que la *visibilitat directa* pel que fa a la potencialitat de la comunicació.

3.4.4 Àmbit de la visibilitat indirecta

Cadena homogènia d'enllaços dirigits. Donada la cadena de visibilitats corresponent a la visibilitat indirecta $A \xrightarrow{*} B^*$, considerem la cadena d'enllaços dirigits que constitueixen una realització d'aquesta cadena. Si tots aquests enllaços dirigits tenen el mateix tipus d'àmbit de validesa (local, de paràmetre o d'atribut) direm que estem davant d'una *cadena homogènia d'enllaços dirigits*.

Visibilitat homogènia. Donada una cadena homogènia d'enllaços dirigits $a : A \rightarrow c_1 : C_1 \rightarrow \dots \rightarrow c_k : C_k \rightarrow b : B$ direm que estem davant una *visibilitat indirecta homogènia* $A \xrightarrow{*} B^*$.

Definició. Visibilitat indirecta d'atribut

Si $A \xrightarrow{*} B$ és una visibilitat homogènia on tots els enllaços dirigits són d'atribut, direm que $A \xrightarrow{*} B$ és una visibilitat indirecta d'atribut.

Definició. Visibilitat indirecta local

Si $A \xrightarrow{*} B$ és una visibilitat homogènia on tots els enllaços dirigits són locals (a $m()$), direm que $A \xrightarrow{*} B$ és una visibilitat indirecta local (a $m()$).

Definició. Visibilitat indirecta de paràmetre

Si $A \xrightarrow{*} B$ és una visibilitat homogènia on tots els enllaços dirigits són de paràmetre (a $m()$), direm que $A \xrightarrow{*} B$ és una visibilitat indirecta de paràmetre (a $m()$).

3.4.5 Multiplicitat de la visibilitat indirecta



Notació. Donada una cadena de visibilitats (no necessàriament homogènia) $A \rightarrow C_1 \rightarrow \dots \rightarrow C_k \rightarrow B^*$, direm que $C_k \rightarrow B^*$ és la *darrera visibilitat* de la visibilitat indirecta $A \rightarrow B^*$.

Definició. Visibilitat indirecta monoavaluada

Si la darrera visibilitat de la visibilitat indirecta és una visibilitat monoavaluada, direm que estem davant d'una visibilitat indirecta monoavaluada.

Definició. Visibilitat indirecta multiavaluada

Tota visibilitat indirecta que no sigui monoavaluada direm que és una visibilitat indirecta multiavaluada.

Cadena de visibilitats directes monoavaluades. La visibilitat indirecta monoavaluada significa l'existència d'una cadena de visibilitats directes monoavaluades; és a dir, l'existència d'una visibilitat monoavaluada (en el sentit que accedim a un sol objecte) però amb la possible existència d'intermediaris.



Notació. Per expressar una visibilitat indirecta monoavaluada entre A i B usarem $A \rightarrow^* B$.

Exemple 33 (Multiplicitat de la visibilitat indirecta) En l'exemple 32 el que tenim és una visibilitat indirecta multiavaluada: $A \rightarrow^* C^*$.

Si l'enllaç dirigit multiavaluat $b \rightarrow C^*$ el canviem per un enllaç dirigit monoavaluat com $b \rightarrow C : C$, llavors el que obtenim és una visibilitat indirecta monoavaluada: $A \rightarrow^* C$.

3.5 Delegació

Definició. Delegació

Direm que un component A delega la responsabilitat $m()$ en el component B quan la realització de la responsabilitat $m()$ es redueix únicament i exclusivament a una petició d'exercici de responsabilitat a través d'una de les visibilitats $A \rightarrow B^*$.

La visibilitat $A \rightarrow B^*$ emprada per la delegació l'anomenarem visibilitat de la delegació.

El mecanisme de la delegació. Per tal d'exercir una responsabilitat $m()$ enviem un missatge $m()$ a un objecte $a : A$. Si l'exercici d'aquesta responsabilitat

és per *delegació* vol dir que l'objecte $a : A$ es limita a emprar l'enllaç dirigit que assegura la *visibilitat de la delegació* per enviar a través d'ell un missatge $m1()$.

Una responsabilitat; diverses operacions. En el mecanisme que acabem d'explicar la responsabilitat inicial assignada a A en forma del missatge $m()$ (i l'operació del mateix nom) s'acaba realitzant com una responsabilitat assignada a B , que associem amb el missatge $m1()$ (i l'operació del mateix nom). En el fons, però, tant $m()$ com $m1()$ poden considerar-se una mateixa responsabilitat (potser sota perspectives diferents). És a dir, una mateixa responsabilitat a A l'exerceix una operació $m()$, mentre que a B l'exerceix una operació $m1()$. El resultat és una responsabilitat exercida per operacions diferents.

Un nom; diferents operacions. Per refermar la idea de delegació sovint s'acostuma a usar el mateix nom tant per l'operació que es delega ($m()$ en la nomenclatura usada en el paràgraf *El mecanisme de la delegació*) com per la operació que rep la delegació ($m1()$ si usem la mateixa nomenclatura). Cal tenir present que es tracta d'operacions diferents, encara que només sigui perquè pertanyen a components diferents.

"Retransmissió" o "reenviament". Sovint la *delegació* es veu com una "retransmissió" o "reenviament" de la petició rebuda. Si bé l'aspecte sintàctic de la delegació³⁵ reforça aquesta visió, només pot ser defensada com una metàfora: l'operació que s'envia, per molt que tingui el mateix nom, no és la mateixa que la l'operació rebuda!!

Excursió. (Implementació) En la delegació el codi de llenguatges OOP sempre té la pinta següent:

```
capçaleraOperacioPerDelegar:
    objecteQueRepLaDelegacio.OperacioQueFaLaTasca;
```

Si usem el criteri de la coincidència de noms en les operacions que intervenen en la delegació, llavors tenim:

```
capçaleraOperacioPerDelegar:
    objecteQueRepLaDelegacio.OperacioPerDelegar;
```

Exemple 34 (Delegació de responsabilitat) Una *Etapa* d'una *Caminada* està formada per dos *Indrets*; l'indret origen i l'indret de destinació. Hem decidit modelitzar això amb dues visibilitats diferents: la visibilitat $\text{origen:Etapa} \rightarrow \text{Indret}$, i la visibilitat $\text{dest:Etapa} \rightarrow \text{Indret}$.

Suposem que els mecanismes d'assignació de responsabilitat ens han portat a introduir l'operació `mostrarInici()` a *Etapa*, que mostra la descripció i la ubicació de l'indret que és inici de l'etapa. Sembla lògic assignar la responsabilitat de mostrar les dades d'un $i : \text{Indret}$ al propi *Indret*. Així, la responsabilitat `mostrarInici()` de l'*Etapa* la deleguem

³⁵Penseu per exemple, en els disgrams de col·laboració o en el codi Java.

en la responsabilitat `mostrar()` de l'Indret corresponent segons la visibilitat de delegació `inici`.

En termes de codi Java tindrem:

```
class Etapa
{
    Indret inici;      // Enllaç de la delegació
    Indret dest;
    .....
    void mostrarInici()
    {
        inici.mostrar(); // Delegació de
                        // mostrarInici()
                        // a mostrar() de
                        // l'Indret obtenible
                        // amb l'enllaç
                        // dirigit inici
    }
}
```

Aquí calen dues operacions amb nom diferent: l'operació externa (`mostrarInici()` engloba en el seu nom quina és la visibilitat de delegació que cal emprar). Una alternativa era passar com a paràmetre la informació necessària per discriminar si es vol mostrar l'inici o destinació de l'etapa. Però llavors abans de fer la delegació caldria interpretar aquest argument. (El resultat tot i que es pot considerar una delegació no s'adapta exactament a la definició donada, que exigeix que l'única tasca realitzada sigui la petició a través de la visibilitat de delegació).

Exemple 35 (Delegació de responsabilitat) Sigui una empresa jerarquitzada en diferents departaments. Una manera de tractar les reclamacions rebudes per l'empresa és delegar-les al departament de reclamacions:

```
class Empresa
{
    Departament reclamacions;
    .....
    void reclamacio(Rec r)      // Rec és una classe
                                // que manté les dades
                                // d'una reclamació
    {
        reclamacions.reclamacio(rec);
    }
}
```

En aquest cas a les operacions involucrades en la delegació els hem donat el mateix nom.

Visibilitat de delegació multiavaluada. La definició de delegació exigeix que en la realització de la responsabilitat delegada es faci únicament i exclusiva una petició a través de la visibilitat de la delegació. En el cas que la visibilitat de la delegació sigui multiavaluada, aquest missatge únic arribarà a tots els objectes destinació de l'enllaç dirigit emprat.

Exemple 36 (Delegació multiavaluada de responsabilitat) En l'exemple 28, \rightarrow presentavem un disseny amb una visibilitat multiavaluada `puntsInterès:GràficaMatemàtica→Punt*`, que permetia relacionar una gràfica amb els seus punts d'interès constituents. En aquest cas el component `GràficaMatemàtica` delega l'operació `visualitzar()` sobre l'operació `visualitzar()` dels `:Punts` que en són els seus constituents (és a dir, usa la visibilitat `puntsInterès` com a visibilitat de la delegació).

Exemple 28. Agregació o no de visibilitats (de component), pàgina 62

Visibilitat de delegació indirecta. La definició de delegació exigeix la presència d'una visibilitat directa: la visibilitat de la delegació. En el cas d'existir una visibilitat indirecta $A \xrightarrow{*} B$ es pot usar la cadena de visibilitat directes que aquesta implica per anar repetint la delegació, des de qui rep la responsabilitat fins a qui realment la realitza: cada element de la cadena té visibilitat directa del següent element de la cadena, i per tant pot delegar-li la responsabilitat que se li ha encomanat, de tal manera que el resultat és com una delegació des de A a B , a través del intermediaris que conformen la cadena de visibilitat emprada.

Exemple 37 (Delegació indirecta) En l'exemple 32, \rightarrow l'objecte a pot delegar en l'objecte b , que al seu torn pot delegar en les diferents realitzacions de C que són destinació de l'enllaç dirigit multiavaluat pertinent. L'efecte global és una delegació de a sobre les diferents realitzacions de C visibles a través de la visibilitat indirecta de delegació $A \xrightarrow{*} C$.

Exemple 32. Visibilitat indirecta, pàgina 66

Cadena de delegacions. En el cas de la delegació a través d'una visibilitat indirecta, la cadena de visibilitats directes que componen aquesta sovint l'anomenem *cadena de delegació*; els missatges que circulen a través d'aquesta cadena constitueixen l'anomenada *cadena de delegacions*.

Exemple 38 (Cadena de delegació) En l'exemple 32, \rightarrow tenim la cadena de delegació $a \rightarrow b \rightarrow C^*$.

Exemple 32. Visibilitat indirecta, pàgina 66

Suposem que a delega la responsabilitat $m()$ a b mitjançant un missatge $m2()$; i el component B exerceix aquesta responsabilitat delegant-la sobre la visibilitat $B \rightarrow C^*$ mitjançant el missatge $m2()$. Llavors direm que la cadena de delegacions necessària per exercir la responsabilitat $m()$ és: $m1()$ (sobre B , a través de $A \rightarrow B$) i $m2()$ (sobre C , a través de $B \rightarrow C^*$).

3.6 Conclusions sobre la visibilitat

Potencialitat. La *visibilitat* és una propietat dels components que expressa la *potencialitat* que les seves realitzacions puguin mantenir determinats enllaços dirigits: si A té visibilitat sobre B les realitzacions de A poden ser origen d'enllaços dirigits que com a destinació tinguin realitzacions de B. Per tant, la *visibilitat* entre els components és la *potencialitat* de col·laboració.

Potencialitat realitzada. L'existència d'una visibilitat exigeix la seva realització: la *potencialitat* que expressa no és hipotètica, sinó que alguns objectes realment mantenen els enllaços dirigits permesos.

Enllaç dirigit. Un enllaç dirigit és la realització d'una visibilitat; n'és una concreció. Per tant els enllaços dirigits només poden existir entre objectes.

Visibilitat d'objectes. Per extensió del vocabulari, si un objecte $a:A$ és l'origen d'un enllaç dirigit que té com a destinació una o més realitzacions de B, direm que l'objecte $a:A$ té *visibilitat* sobre les realitzacions concretes de B que són destinació de l'enllaç.

Una moneda amb dues cares. Així, *visibilitat* i *enllaços dirigits* són les dues cares d'una mateixa moneda: *potencialitat* i *realització*; *generalització* i *concreció*.

Multiavaluació dels enllaços. Els enllaços dirigits poden tenir més d'una destinació; en aquest cas parlem d'*enllaços dirigits multiavaluats*, en contraposició als *enllaços dirigits simples* o *monoavaluats* que només tenen una destinació. Els *enllaços dirigits generalitzats* consideren els *enllaços dirigits simples* o *monoavaluats* com un cas particular dels *enllaços dirigits multiavaluats*.

Multiavaluació de la visibilitat. La generalització (en termes de *potencialitat*) de la multiavaluació dels enllaços porta a la *visibilitat multiavaluada*. La *visibilitat generalitzada* considera la *visibilitat monoavaluada* (aquella que es realitza amb enllaços dirigits monoavaluats) un cas particular de la *visibilitat multiavaluada*.

Col·laboració i visibilitat multiavaluada. La *visibilitat multiavaluada* permet usar un mateix enllaç per col·laborar amb múltiples objectes, és a dir, permet que un mateix missatge arribi a més d'un objecte.³⁶

Indirecció en els enllaços dirigits. L'existència d'una seqüència d'enllaços dirigits enllaçables (la destinació d'un com a origen d'un altre) permet parlar de *cadena d'enllaços dirigits* o d'*enllaços dirigits indirectes*: l'accés a la destinació (o destinacions) de l'enllaç es fa a través de múltiples intermediaris.

³⁶A nivell de disseny es tracta d'un sol missatge, tot i que segurament en implementació el que es farà és generar tantes còpies del missatge com destinacions tingui l'enllaç dirigit emprat. \rightarrow

Visibilitat indirecta. La generalització (en termes de potencialitat) dels *enllaços dirigits indirectes* permet parlar de *visibilitat indirecta*. A efectes comunicatius la *visibilitat indirecta* i els *enllaços dirigits indirectes* es comporten exactament igual que la *visibilitat (directa)* i els *enllaços dirigits (directes)*.

Cadenes de visibilitat. La visibilitat indirecta significa l'existència de *cadena d'enllaços dirigits*; i per tant l'existència d'una *cadena de visibilitats*. Però, si bé l'enllaçabilitat d'un conjunt d'enllaços dirigits significa l'existència d'una *cadena d'enllaços dirigits* (i per tant de la *cadena de visibilitats* corresponent), l'enllaçabilitat d'un conjunt de visibilitats poca cosa diu sobre l'existència d'una *cadena de visibilitats*. Per poder parlar de *cadena de visibilitats* cal assegurar l'existència de *cadena d'enllaços dirigits* que en siguin una realització.

Col·laboració i visibilitat indirecta. La *visibilitat indirecta* permet la col·laboració a través d'intermediaris. Així, pel que fa a la potencialitat d'enviar missatges la visibilitat indirecta es comporta exactament igual que la visibilitat directa

Sobre la notació. La notació usa la mateixa simbologia per expressar els enllaços dirigits i la visibilitat: d'aquesta manera es reforça la idea que són les dues cares d'una mateixa moneda. Per altra banda s'usa l'asterisc (*) per indicar iteració. La iteració sobre la destinació (de l'enllaç o la visibilitat) indica *multiavaluació*; la iteració sobre el símbol d'enllaç o de visibilitat (\rightarrow) indica *indirecció*. Finalment, no es distingeix notacionalment entre la multiavaluació i la generalització: l'asterisc, llegit com a iteració obligada indica multiavaluació; llegit com a iteració potestativa indica generalització.

Col·laboració per delegació. Un cas particular de col·laboració és la delegació. En ell s'usa una visibilitat (monoavaluada o multiavaluada; directa o indirecta) per fer una sola petició que resolgui la tasca que se'ns ha assignat. (És a dir, enlloc d'exercir la responsabilitat, demanem a algú altre que ho faci per nosaltres).

4 Col·laboració

Tipus de col·laboracions. Donat un objecte $a:A$ i una visibilitat $A \rightarrow B^*$ els diferents tipus de col·laboració possible són la col·laboració a través de l'enllaç; la col·laboració via *selecció*; i la col·laboració de *grup*. Tot seguit les analitzem.

Col·laboració d'enllaç. La *col·laboració d'enllaç* és el mecanisme de col·laboració usat fins ara: l'objecte a envia un missatge $m()$ a través de l'enllaç dirigit $a \rightarrow B^*$; el resultat és un mateix missatge $m()$ arriba a tots els objectes que són destinació de l'enllaç dirigit $a \rightarrow B^*$.

Col·laboració amb selecció. En la *col·laboració amb selecció* l'objecte $a:A$ vol enviar un missatge $m()$ a un dels múltiples objectes que són destinació de l'enllaç dirigit $a \rightarrow B^*$. Per fer-ho ha de fer una *selecció* sobre la destinació

de l'enllaç dirigit $a \rightarrow B^*$ per determinar quin és l'objecte a qui es vol enviar el missatge $m()$.

Exemple 39 (Col·laboració amb selecció)

En l'exemple 27, \rightarrow (que fa referència a l'exemple 22, pàgina 58) tenim una visibilitat $Etapa \rightarrow Indret^2$. Suposem que $Indret$ respon al missatge `mostraDescripcio()` mostrant les dades de l'indret. Llavors, si volem mostrar les dades de l'origen de d'una $e:Etapa$ caldrà usar la visibilitat $e \rightarrow Indret^2$ per recuperar l'objecte origen $or:Indret$, i tot seguit enviar-li el missatge `mostraDescripcio()`.

Alternativament, en enviar el missatge `mostraDescripcio()` es pot indicar d'alguna manera³⁷ quin dels objectes destinació de l'enllaç és el receptor del missatge. Per exemple, en enviar `mostraDescripcio()*Sel(1)` per l'enllaç dirigit $e \rightarrow Indret^2$ volem expressar que el missatge `mostraDescripcio()` només ha d'arribar al primer objecte dels que conformen la destinació de l'enllaç dirigit en qüestió.

Col·laboració amb el grup. En la col·laboració amb el grup l'objecte $a:A$ envia un missatge $m()$ al grup d'objectes que són destinació de l'enllaç; però vol que el missatge arribi al conjunt d'objectes com a tal, no pas a cadascun dels objectes de manera individual.

Exemple 40 (Col·laboració d'agregació)

Suposem una visibilitat singulars $Etapa \rightarrow Indret^*$ que permet accedir a tots els indrets singulars que constitueixen una etapa. Suposem que una de les operacions que ofereix $Indret$ és `esCastell` que indica si l'indret és un castell o no.

Quin és el mecanisme per esbrinar si una $e:Etapa$ passa per un castell o no? Una possibilitat és enviar el missatge `esCastell` al grup format per tots els objectes que són destinació de l'enllaç dirigit $e \rightarrow Indret^*$. (Assumim que un missatge booleà sobre un grup retorna cert si i només si hi ha algun element del grup que respon afirmativament al missatge).

En aquest exemple la col·laboració de grup apareix com un mecanisme que agrega les respostes resultant de diferents col·laboracions de selecció.

La visió de la col·laboració amb el grup com una agregació de seleccions és un artífici sintàctic simplificador, però no aporta res de nou a la col·laboració d'enllaç.

³⁷L'ús d'un paràmetre extra per expressar la selecció converteix la col·laboració de selecció en una col·laboració de grup: el missatge el rep el grup, qui l'interpreta i envia el missatge corresponent a l'objecte pertinent. El mecanisme és molt similar al dels *multiobjectes*, analitzat més endavant. \rightarrow En la col·laboració per selecció pròpiament dita, el missatge té els seus propis arguments; a part, en l'emissió del missatge el mecanisme d'emissió (i no pas el missatge) requereix com a argument la informació necessària per tal de poder fer la selecció (vegeu l'exemple en aquest mateix paràgraf).

3.5 Modelització de les associacions del MC

J.M. Merenciano

Com a operació d'agregació de respostes en aquest exemple hem usat l'operació or lògica. Altres operacions possibles són, per exemple, la and lògica, i el sumatori o productori de valors numèrics.

Exemple 41 (Col·laboració amb el grup)

Suposem que volem afegir un nou $i : \text{Indret}$ com a destinació de l'enllaç dirigit singulars: $e \rightarrow \text{Indret}^$. Com ho fem?*

En aquest cas no té sentit fer diferents seleccions i agregar o compondre d'alguna manera els resultats individuals. No hi ha més remei que enviar un missatge $\text{afegir}(i : \text{Indret})$ a tot el grup.

Aquesta és l'autèntica col·laboració amb el grup: la col·laboració obtinguda no es pot aconseguir amb combinacions de col·laboracions dels altres tipus.

5 Modelització de les associacions del MC

5.1 Les associacions en el MC

Semàntica de les associacions de MC. En el MC una associació X entre dos conceptes A i B és una abstracció que agrupa tots els enllaços entre realitzacions de A i B que es corresponen a realitzacions de la propietat X .

Estaticitat de les associacions del MC. L'associació X és una propietat estàtica (per això està en el MC) que ens diu que en tot moment tota realització de A està enllaçada amb alguna realització de B (i viceversa).³⁸ Els enllaços³⁹ com a tals poden ser dinàmics (variants en el temps) o no.

5.2 Model de les associacions en el disseny

En el disseny, i pel principi de la *Modelització continua*, tenim dos components^{40, →} A i B que modelen els conceptes del mateix nom. El problema que cal resoldre ara és com modelar l'associació X . Si apliquem novament el principi de la *Modelització continua*, l'associació del MC s'ha de modelitzar amb una propietat estàtica que digui que en tot moment tota realització del component A està enllaçada amb alguna realització del component B (i viceversa).

3.1.4. Enllaços dirigits, pàgina 40

5.2.1 Associacions del MC i visibilitat d'atribut

Semàntica de la visibilitat d'atribut. La visibilitat d'atribut és el mecanisme que permet modelitzar les associacions del MC. Si A té visibilitat d'atribut

³⁸Aquesta afirmació ha d'anar matisada per les multiplicitats i obligatorietats o optativitats de l'associació.

³⁹Aquí el terme *enllaç* és un terme de l'especificació: és el lligam entre realitzacions de conceptes; és la realització d'una associació entre conceptes.

⁴⁰A l'apartat 12., pàgina 42, de 3.1.4. Enllaços dirigits, pàgina 40, s'han presentat els components com el model o correlat software dels conceptes.

sobre B vol dir que tota realització $a : A$ té la potencialitat de tenir un enllaç dirigit d'atribut $a \rightarrow B^*$. Per tractar-se d'un enllaç dirigit d'atribut la seva validesa és independent de l'operació que estigui realitzant $a : A$; en aquest sentit és una *propietat estàtica* de $a : A$.



Notació. Per indicar una visibilitat d'atribut $A \rightarrow B^*$ usarem $A \xrightarrow{atr} B^*$.

Visibilitat d'atribut i associació. Sigui una associació *assoc* entre els conceptes A i B , amb multiplicitat $n - m$. La presència d'una visibilitat d'atribut $assoc : A \xrightarrow{atr} B^*$ significa que tota realització del component A (és a dir, tota realització del correlat software del concepte A) té potencialitat per mantenir un enllaç dirigit "estàtic" (és un enllaç dirigit d'atribut, i per tant vàlid independentment del context dinàmic considerat) cap a m realitzacions del correlat software del concepte B . Per tant, la visibilitat $A \xrightarrow{atr} B^*$ és un model per a l'associació *assoc* entre A i B .

Model de les associacions. El model software de les associacions són les *visibilitats d'atribut*. Les associacions són l'abstracció dels enllaços entre les realitzacions dels components; el model software d'aquests enllaços són els *enllaços dirigits d'atribut*.

Semàntica dels enllaços dirigits. En la discussió sobre els enllaços dirigits multiavaluats[→] s'ha introduït informalment la idea de semàntica dels enllaços dirigits. En el cas de la visibilitat d'atribut ara estem en condicions de donar una definició més precisa: la semàntica dels enllaços dirigits corresponents a una visibilitat d'atribut és la semàntica de l'associació del qual la visibilitat n'és model.

Múltiples models de les associacions. Cal observar com si bé l'associació *assoc* entre A i B es pot modelitzar amb la visibilitat d'atribut $A \xrightarrow{atr} B^*$, també admet com a model la visibilitat $B \xrightarrow{atr} A^*$ (o fins i tot l'ús d'ambdues visibilitats simultàniament).

Excursió. (Implementació) En llenguatges com el C++ o el Java, els *components* s'implementen amb *classes*; i la *visibilitat d'atribut* s'implementa mitjançant atributs de les classes.

En concret, la visibilitat d'atribut $A \xrightarrow{atr} B^*$ s'implementa amb un atribut de tipus B (en el cas de la visibilitat monoavaluada)⁴¹ dins la classe A . El nom de l'atribut, en virtut del principi de la *Modelització continua*, coincideix amb el nom del *rol* que s'ha assignat a l'extrem B de l'associació *assoc* que modelitza la visibilitat d'atribut $A \xrightarrow{atr} B^*$ considerada. En cas de no haver donat nom al rol, s'usa el nom de la pròpia associació.

⁴¹La implementació de les visibilitats multiavaluades depen de si el llenguatge admet els multiojectes o no. En el cas del C++ o el Java aquests no s'admeten directament, i cal usar unes estructures intermèdies. A l'apartat 5.4.Incursió en la implementació, pàgina 97[→], ho analitzem amb deteniment.

```
class A
{
    B assoc;
}
```

Direccionalitat dels enllaços. Les associacions representen *enllaços bidireccionals*. La visibilitat d'atribut imposa un sentit a l'enllaç (el sentit de la visibilitat). De cara a l'objectiu d'expressar en el model software les propietats de les associacions del model conceptual, qualsevol orientació dels enllaços és vàlida.

La responsabilitat en l'orientació dels enllaços. Correspon al disseny decidir quin és el sentit de visibilitat més adient: el sentit dels enllaços és el que exigeixen les necessitats de col·laboració entre els diferents objectes.

- En el disseny una associació de A a B es modelitza amb una visibilitat d'atribut $A \xrightarrow{atr} B^*$ (o a la inversa)
- La modelització d'una associació implica convertir⁴² els enllaços bidireccionals en *enllaços dirigits*
- El sentit concret de l'orientació donada als enllaços dirigits el fixen les necessitats de col·laboració del disseny

(13)

⁴²Pròpiament no hi ha cap conversió o orientació de les associacions, ja que la modelització implica canviar el domini: passem del domini del problema al domini de la solució.

Principis i definicions del capítol

Definicions

Cadena d'enllaços dirigits, 63
Context d'emissió d'un missatge, 42
Context de localitat, 46

Delegació, 68

Enllaç dirigit, 40
Enllaç dirigit (generalitzat), 59
Enllaç dirigit d'atribut, 45
Enllaç dirigit de paràmetre, 48
Enllaç dirigit local, 46
Enllaç dirigit monoavaluat o simple, 57
Enllaç dirigit multiavaluat, 57
Enllaços dirigits indirectes, 62

Objecte, 36

Visibilitat indirecta d'atribut, 67
Visibilitat indirecta de paràmetre, 67
Visibilitat (directa) d'un objecte, 52
Visibilitat d'atribut, 53, 56
Visibilitat de paràmetre, 53, 56
Visibilitat entre components, 54
Visibilitat generalitzada (d'objecte), 61
Visibilitat generalitzada (d'un component), 62
Visibilitat indirecta, 65
Visibilitat indirecta local, 67
Visibilitat indirecta monoavaluada, 68
Visibilitat indirecta multiavaluada, 68
Visibilitat local, 53, 57
Visibilitat multiavaluada (d'un component), 61
Visibilitat multiavaluada (d'un objecte), 60

Capítol 4

Models en el disseny

1	El model col·laboratiu	81
2	Model de components (MComp)	81
2.1	Expressivitat i contingut de MComp	81
2.2	Dependències	82
3	El perill de l'enginyeria inversa	83
4	Consistència dels models	84
5	Construcció del MComp	85
5.1	Validació	85
5.2	Síntesi	86

Contingut detallat del capítol 4

1 El model col·laboratiu	81
Model col·laboratiu i model de comportament	81
Expressivitat del model col·laboratiu	81
Elements del model col·laboratiu	81
Excursió. (UML)	81
2 Model de components (MComp)	81
2.1 Expressivitat i contingut de MComp	81
Model de components (MComp) i model conceptual (MC) .	81
Expressivitat de MComp com a model de MC	81
Correlats software dels elements de MC	81
Excursió. (UML)	81
Potencialitat de comportament	81
Contingut de MComp	81
2.2 Dependències	82
Violació d' <i>Espill</i>	82
Dependències	82
Conseqüències de les dependències	82
Explicitació de les dependències	82
Visibilitat d'atribut i les altres dependències	82
Excursió. (UML)	82
Excursió. (Visibilitat dinàmica i visibilitat estàtica)	83
3 El perill de l'enginyeria inversa	83
Enginyeria inversa	83
Violació d' <i>Espill</i>	83
Dissenys diferents	84
Fragilitat de l'enginyeria inversa	84
4 Consistència dels models	84
Propietats de consistència	84
Propietats generals de consistència	84
Model mínim	85
Necessitat d'explicitar les dependències	85
5 Construcció del MComp	85
5.1 Validació	85
Model de partida	85
Errors en el model de partida	85
Tasques del disseny de validació	85
Validació, orientació i dependències	86
Supressió	86
5.2 Síntesi	86

1 El model col·laboratiu

Model col·laboratiu i model de comportament. El *model col·laboratiu* forma part del disseny; és el correlat del *model de comportament* de l'especificació.

Expressivitat del model col·laboratiu. En el *model col·laboratiu* cal expressar les funcionalitats especificades del sistema en termes de *col·laboració entre els objectes*.

Elements del model col·laboratiu. En el *model col·laboratiu* apareixen els *objectes* i els *missatges* que s'envien entre ells. L'emissió d'un missatge exigeix visibilitat del receptor per part de l'emissor, i per tant l'ús d'un *enllaç dirigit*.

Excursió. (UML) En UML usarem els *diagrames d'interacció* (i d'aquests, per comoditat, generalment usarem els *diagrames de comunicació*). El model col·laboratiu es correspondrà amb el conjunt de tots els diagrames d'interacció desenvolupats.

En els diagrames d'interacció els enllaços dirigits emprats no s'acostumen a explicitar. Només ho farem quan hi hagi més d'un enllaç dirigit possible (com per exemple en el cas de les Etapes amb dos Indrets singulars associats: origen i destinació).

2 Model de components (MComp)

2.1 Expressivitat i contingut de MComp

Model de components (MComp) i model conceptual (MC). El *model de components (MComp)* forma part del disseny; és el correlat del *model conceptual (MC)* de l'especificació.

Expressivitat de MComp com a model de MC. En el *model de components* cal fer aparèixer el *model* dels conceptes, i el *model* de les associacions del MC que siguin rellevants de cara a la construcció de la solució.¹

Correlats software dels elements de MC. El *component* és el model (o correlat software) del *concepte*.[→] La *visibilitat d'atribut* és el model (o correlat software) d'una *associació*.[→]

Excursió. (UML) En UML els *components* s'expressen amb rectangles (a l'igual que els conceptes); les *visibilitats d'atributs* amb arcs dirigits (les associacions es representen amb arcs neutres, sense direcció).

Potencialitat de comportament. El *model de components (MComp)* conté els *components* considerats. Per tant sembla el lloc idoni per expressar les potencialitats col·laboratives d'aquests; que en definitiva són les que s'exerceixen en el *model col·laboratiu*. Aquestes potencialitats són les *operacions* ofertes per cada *component*.

¹El principi *Espill* exigeix que tot element MComp es correspongui amb un element de MC, però permet que elements de MC no apareguin en el seu model MComp.

3.1.Elements estructurals, pàgina 36

3.5.2.Model de les associacions en el disseny, pàgina 75

Contingut de MComp. De tot plegat resulta que *MComp* ha d'expressar els *components* considerats, les *operacions* ofertes per cada component, i les *visibilitats* entre els diferents components.

2.2 Dependències

Violació d'Espill. Més amunt,[→] hem vist com a *MComp* s'havien d'expressar les *visibilitats* entre els diferents components. També hem vist que la visibilitat d'atribut és el model software de les associacions del MC.[→] Quin és però el paper de les visibilitats locals i de paràmetre? En fer-les aparèixer a *MComp* violem el principi de l'Espill!!

Dependències. Les visibilitats expressen algun tipus de *dependència* entre els components: si existeix la visibilitat $A \rightarrow B^*$ llavors la necessitat² del component A de conèixer B fa que A sigui *dependent* de B.

Conseqüències de les dependències. L'existència d'una dependència té força conseqüències. Sovint es fa difícil entendre un component si no s'entenen les interrelacions que té amb els components dels quals depèn; d'aquí que hi hagi principis de disseny que pretenuguin minimitzar la dependència;³ o hi hagi mètodes de desenvolupament que fixen l'ordre de desenvolupament en funció del grau de dependència. A més, a nivell d'implementació les dependències poden implicar necessitats de recompilació cada cop que es toca algun element del qual en depenem.

Explicitació de les dependències. Així les dependències afecten la comprensió del disseny, el propi disseny, i el de desenvolupament, sense oblidar que tenen conseqüències a nivell d'implementació. D'aquí la necessitat de fer aparèixer a *MComp* totes les *dependències* del disseny construït.

Visibilitat d'atribut i les altres dependències. La visibilitat d'atribut és una dependència que modelitza les associacions de MC. Les visibilitats locals i de paràmetre són dependències que violen *Espill*: les volem en el model software però no es corresponen a cap element del model conceptual.⁴

Excursió. (UML) En UML les *visibilitats d'atribut* s'expressen amb arcs dirigits continus. Les visibilitats locals i de paràmetre en UML reben el nom de *dependència*, i s'expressen amb arcs dirigits discontinus. Per indicar si la dependència és local o de paràmetre es poden usar *estereotips*.⁵

²La visibilitat és opcionalitat de coneixement de cara als objectes. Però de cara al component origen de la visibilitat és una obligació: només gràcies a què A coneix B pot oferir els mecanismes pertinents a les seves realitzacions per tal que tinguin coneixement i accés a realitzacions del component B.

³Principi de *Baix acoblament*, per exemple.

⁴Si més no de forma explícita.

⁵Per simplificar, un *estereotip* el podem considerar com una etiqueta predefinida per l'usuari.

Així, la notació UML distingeix entre les visibilitats que es corresponen a algun element del problema, de les que són pura ficció per les necessitats del software.

Excursió. (Visibilitat dinàmica i visibilitat estàtica) MComp és un model estàtic; en ell s'hi representen les *visibilitats estàtiques*: dependències i visibilitats d'atribut.

La *visibilitat estàtica d'atribut* $A \xrightarrow{atr} B^*$ diu que en tot moment cada realització $a : A$ pot tenir visibilitat (generalitzada) sobre realitzacions de B .

La *visibilitat estàtica per dependència* d'un component A cap un component B diu que donada una realització $a : A$ qualsevol, en algun moment pot tenir necessitat de tenir visibilitat sobre realitzacions de B . La *visibilitat estàtica per dependència* expressa la possibilitat d'una visibilitat en un moment donat; i per tant d'alguna manera expressa les necessitats de coneixement d'uns components sobre els altres.

La *visibilitat dinàmica* és la visibilitat concreta que un objecte concret té en un moment determinat.

La *visibilitat estàtica d'atribut* expressa quins enllaços són possibles, i quines assumpcions d'existència es poden fer; la *visibilitat estàtica per dependència* expressa quines possibilitats d'enllaç dirigit poden ser necessàries temporalment; la *visibilitat dinàmica* expressa exactament quins enllaços dirigits hi ha en un moment determinat.

La *visibilitat dinàmica* es pot extreure dels *diagrames d'interacció*. La *visibilitat estàtica* és d'alguna manera l'abstracció de la *visibilitat dinàmica*, i hom espera que sigui coherent amb el MC; en concret que la *visibilitat estàtica d'atribut* sigui el model de les associacions.

3 El perill de l'enginyeria inversa

Enginyeria inversa. Suposem el mecanisme de simulació dels multiobjectes en Java:^{6,→} cada multiobjecte requereix la introducció d'un atribut d'un tipus `Collection`. Si pensem en termes d'implementació, a l'hora de dissenyar podem aplicar el principi de l'*Espill* i postular l'existència d'un component `Multiobjecte` que és la "realitat" modelada per la `Collection`. Hem fet *Enginyeria inversa*: hem aplicat la modelització des del model (implementació) a la "realitat" (disseny).

5.4.Incursió en la implementació, pàgina 97

Violació d'*Espill*. L'enginyeria inversa realitzada ens porta a un model que viola el principi de l'*Espill*: el component introduït no és model de cap concepte!!

Exemple 42 (L'enginyeria inversa pot violar *Espill*) En l'exemple 50, → l'ús de l'enginyeria inversa fa aparèixer un component `List` a `MComp`.

Exemple 50.Emissió gestionada, pàgina 99

En l'exemple 50, pàgina 99, l'ús de l'enginyeria inversa fa aparèixer

⁶A l'apartat 5.4.Incursió en la implementació, pàgina 97, s'ha introduït en forma d'exemples.

■ un component $List$ i un component $Gestor$ a $MComp$.

Dissenys diferents. L'enginyeria inversa significa que termes o decisions d'implementació poden influir en el disseny. Una conseqüència és la violació d'*Espill*; una altra conseqüència força més greu és que diferents implementacions signifiquen dissenys diferents.

Exemple 50. Emissió gestionada, pàgina 99

Exemple 51. Recepció gestionada, pàgina 99

Exemple 43 (Llibertat en la implementació) L'ús de l'enginyeria inversa a l'exemple 50 \rightarrow dóna com a conseqüència la visibilitat $A \xrightarrow{atr} List \xrightarrow{atr} B^*$.

L'ús de l'enginyeria inversa a l'exemple 51 \rightarrow dóna com a conseqüència la visibilitat $A \xrightarrow{atr} Gestor \xrightarrow{atr} List \xrightarrow{atr} B^*$.

Els dos dissenys són diferents en els components i en les visibilitats. I de fet l'únic que ha variat és la decisió de qui gestiona els missatges, si el receptor o l'emissor.

Si no fem enginyeria inversa hi ha un sol model: $A \xrightarrow{atr} B^*$. A nivell d'implementació tenim llibertat per escollir com simulem el multiobjecte i con gestionem els missatges: la implementació no afecta el disseny, que romà invariable.

Fragilitat de l'enginyeria inversa. En usar l'enginyeria inversa el disseny resultant és molt fràgil: qualsevol petit canvi d'implementació el pot afectar de ple. Aquesta fragilitat afecta directament la mantenibilitat i extensibilitat del sistema.

4 Consistència dels models

Propietats de consistència. El model col·laboratiu i el model de components han de ser *consistents*. En concret:

1. **Objectes vàlids.** Tot objecte que apareix en el model col·laboratiu és realització d'un component de $MComp$
2. **Enllaços vàlids.** Tot enllaç usat en el model col·laboratiu es correspon a una visibilitat de les expressades a $MComp$ (sigui en forma de visibilitat d'atribut o de dependència)
3. **Components útils.** Tot component de $MComp$ té alguna realització que participa en una col·laboració
4. **Visibilitats útils.** Tota visibilitat (d'atribut, local o de paràmetre) que apareix a $MComp$ és usada en algun moment en el model col·laboratiu per enviar algun missatge.

Propietats generals de consistència. Les propietats acabades d'exposar són un cas particular de les següents condicions [generals de consistència](#):

Condicions de consistència i minimalitat

- **Abstracció necessària.** Tot element concret ha de ser realització d'una abstracció.
- **Estructura mínima.** Tota abstracció ha de tenir una concreció que la realitzi.
- **Comportament induït.** El comportament de les realitzacions és aquell que les corresponents abstraccions permeten

(14)

Model mínim. Les propietats de consistència diuen que a MComp només apareixen els components i visibilitats necessàries i suficients per a les col·laboracions expressades. En aquest sentit el MComp construït és mínim en relació a les col·laboracions proposades.

Necessitat d'explicitar les dependències. El desig de la consistència i minimalitat dels models és un argument fort a favor de la presència explícita de les dependències a MComp, i la corresponent violació d'*Espill*.⁷ → En cas de no explicitar les dependències, tota col·laboració a través d'un enllaç dirigit local o de paràmetre seria una col·laboració que no es podria contrastar o validar en termes dels permisos que MComp otorga als diferents objectes.

4.2.2.Dependències, pàgina 82

5 Construcció del MComp

Tenim dos mecanismes: *validació* i *síntesi*.

5.1 Validació

Model de partida. En virtut de la *Modelització continua* partim d'un pseudo model de components, isomorf al MC.

Errors en el model de partida. Per construir el MComp inicial, els conceptes de MC els fem correspondre amb components homònims, i les associacions amb les visibilitats d'atribut pertinents. El problema és que en la modelització de les associacions qualsevol direccionalitat en la visibilitat d'atribut és vàlida, i és el disseny qui s'ha de decidir per una de les possibilitats. Com que encara no hem començat a dissenyar no podem prendre cap decisió. El que fem és mantenir al MComp inicial les "associacions", i a mesura que anem avançant les anirem "convertint" en visibilitats d'atribut. D'aquesta manera el model inicial de partida és un model erroni (les "associacions" no són cap element vàlid a MComp).

⁷A l'apartat 4.2.2.Dependències, pàgina 82, hem vist d'altres arguments.

Tasques del disseny de validació. El disseny ha de:

- Validar la necessitat dels components i les “associacions” presents
- Determinar el sentit de les “associacions” validades, i d’aquesta manera convertir-les en visibilitats d’atribut
- Determinar les dependències

Validació, orientació i dependències. Els components necessaris són aquells als qui el disseny assigna responsabilitats. La recomposició col·laborativa proposada en el disseny exigeix determinades visibilitats d’atribut; les “associacions” necessàries així com la seva orientació són conseqüència directa d’aquestes visibilitats d’atributs. Les dependències reflecteixen les visibilitats necessàries per a la recomposició col·laborativa que no són visibilitats d’atribut.

Supressió. Finalment, els components que no han rebut responsabilitat, i les “associacions” que no es corresponen a visibilitats d’atribut, s’han d’esborrar de MComp.

5.2 Síntesi

Construïm MComp de manera incremental. En la tasca d’assignació de responsabilitats decidim si es poden assignar les responsabilitats als components presents al nostre model actual, o si per contra ens cal afegir nous components. En aquest darrer cas, i fent ús del principi de la *Modelització continua*, introduïm un nou component que sigui correlat d’algun dels conceptes del MC. Tant en un cas com en un altre, en la tasca de recomposició col·laborativa afegim les “associacions orientades” i les dependències que es corresponguin a les visibilitats emprades.

Capítol 5

Multiobjectes

1	Presentació dels <i>multiobjectes</i>	90
1.1	Què és un <i>multiobjecte</i>	90
1.2	Operacions sobre un <i>multiobjecte</i>	90
1.3	Visibilitat i <i>multiobjectes</i>	91
2	Col·laboració i <i>multiobjectes</i>	92
2.1	Mecanisme homogeni de col·laboració	92
2.2	<i>Multiobjectes</i> en una cadena de col·laboracions	94
3	El <i>multiobjecte</i> com a artifici	96
4	Incursió en la implementació	97
5	MOMO: Un modisme pels <i>multiobjectes</i>	100
	Principis	102
	Definicions	102

Contingut detallat del capítol 5

1 Presentació dels <i>multiobjectes</i>	90
1.1 Què és un <i>multiobjecte</i>	90
La dualitat unitat-col·lectiu	90
Utilitat dels <i>multiobjectes</i>	90
1.2 Operacions sobre un <i>multiobjecte</i>	90
Operacions sobre un <i>multiobjecte</i>	90
Contracte de les operacions sobre els <i>multiobjectes</i>	90
1.3 Visibilitat i <i>multiobjectes</i>	91
Multiobjectes i visibilitat multiavaluada d'objectes	91
Dualitat unitat-col·lectiu en la notació (1)	91
Multiobjectes i visibilitat multiavaluada de components	92
Dualitat unitat-col·lectiu en la notació (2)	92
2 Col·laboració i <i>multiobjectes</i>	92
2.1 Mecanisme homogeni de col·laboració	92
Homogeneïtat en les col·laboracions	92
Dos tipus d'operacions	92
Col·laboració d'enllaç	92
El <i>multiobjecte</i> com a "intèrpret" d'un missatge	92
Excursió. (Aventatges del <i>multiobjecte</i>)	93
Col·laboració amb selecció	93
Col·laboració amb el grup	93
Excursió. (UML)	94
El mecanisme de col·laboració	94
2.2 <i>Multiobjectes</i> en una cadena de col·laboracions	94
Context de l'anàlisi	94
Objectiu de l'anàlisi	94
La barrera del <i>multiobjecte</i>	94
Traspàs de la barrera del <i>multiobjecte</i>	94
Múltiples enllaços dirigits	95
Transparència del <i>multiobjecte</i>	95
Multiobjectes i visibilitat indirecta	95
3 El <i>multiobjecte</i> com a artifici	96
El <i>multiobjecte</i> vist com un objecte	96
Actuació d'un <i>multiobjecte</i> davant la recepció d'un missatge	96
El <i>multiobjecte</i> com a mecanisme de col·laboració	97
El <i>multiobjecte</i> no és la realització de cap component	97
Multiobjectes i model de components	97
4 Incursió en la implementació	97
Multiobjectes no suportats	97
Simulació dels <i>multiobjectes</i>	97
Operacions sobre els <i>multiobjectes</i> simulats	98
Necessitat de gestionar els missatges sobre <i>multiobjectes</i>	98
Dues gestions possibles	98

Emissió gestionada	98
Recepció gestionada	99
Implementació d'un multiobjecte	100
5 MOMO: Un modisme pels multiobjectes	100
Ús dels modismes	100
Multiobjectes i modismes	100
<i>MOMO: Modisme de l'Objecte MultiObjecte</i>	100
El modisme <i>MOMO</i>	100
Objecte sense component	100
<i>MOMO</i> i les necessitats de creació	100
Responsabilitats sobre els multiobjectes	101
Ús del modisme <i>MOMO</i>	101
Recomanació sobre l'ús de <i>MOMO</i>	101
Principis	102
Definicions	102

1 Presentació dels *multiobjectes*

1.1 Què és un *multiobjecte*

La dualitat unitat-col·lectiu. Les col·laboracions per *selecció* o de *grup*[→] exigeixen veure com una unitat el conjunt d'objectes que constitueixen la destinació d'un enllaç dirigit. Però alhora això no ha d'ocultar els diferents objectes que constitueixen el conjunt: altrament ni la selecció ni l'agregació serien possibles. És a dir, interessa, poder veure la destinació d'un enllaç dirigit com una sola unitat, o com el col·lectiu dels objectes que la conformen.

Utilitat dels multiobjectes. Una manera de tractar amb la *dualitat unitat-col·lectiu* és mitjançant el concepte dels *multiobjectes*:

Definició. Multiobjecte

Un multiobjecte és un artifici que ens permet veure un conjunt d'objectes com si d'un sol objecte agrupador es tractés.



Notació. Usarem $a : A^*$ per expressar que a és un *multiobjecte* de realitzacions del component A .

1.2 Operacions sobre un *multiobjecte*

Operacions sobre un multiobjecte. Per tal de treballar amb els *multiobjectes* assumim que aquests sempre ofereixen com a mínim una operació de selecció d'un objecte del multiobjecte (`find()`), i una operació que permet afegir objectes a un multiobjecte ja existent (`add()`).¹

Contracte de les operacions sobre els multiobjectes. Tot seguit presentem el contracte de les operacions que assumim que ofereix tot *multiobjecte*. L'especificació que donem és volgutament molt general, i segurament caldrà adaptar-la a cada cas concret.

Contracte *find(id) : <e:Elem>*

Paràmetres

- *clau*: Identificador de l'objecte que es vol seleccionar

Retorna

¹Altres operacions poden ser també necessàries o útils. Per exemple, la modificació o supressió d'un objecte del multiobjecte; mecanismes per iterar sobre tots els objectes del multiobjecte; una operació que ens digui quants objectes conté el multiobjecte; etc. Aquí, per simplificar, hem preferit limitar al màxim les operacions disponibles. Si en el disseny hom observa la necessitat d'una altra operació sempre la podrem afegir.

5.1 Presentació dels *multiobjectes*

J.M. Merenciano

- *e*: Objecte dins del multiobjecte que s'identifica amb la *clau* o *identificador* *id* donat. En cas que no hi hagi cap objecte amb aquestes característiques retorna el valor `null`

PRE

1. Cert

POST

1. Sigui $e : \text{Elem}$ l'objecte retornat. Llavors $e.id = id$ o $e = \text{null}$. (Assumim un atribut *id* que manté l'identificador de l'objecte)
2. $e = \text{null}$ si i només si no existeix dins del multiobjecte cap objecte *e* amb $e.id = id$

Contracte ***add*** (*e* : *Elem*)

Paràmetres

- *elem*: Objecte que es vol afegir al multiobjecte

PRE

1. *Homogeneïtat*. Tots els objectes presents dins del multiobjecte són realitzacions del component *Elem*
2. *Unicitat*. Dins del multiobjecte no existeix cap objecte *x* amb $x.id = e.id$. (Assumim un atribut *id* que manté l'identificador de l'objecte)

POST

1. El multiobjecte conté *e*. (Formalment: $\text{find}(e.id) = e$)
2. Els objectes que contenia el multiobjecte encara els conté.

1.3 Visibilitat i multiobjectes

Multiobjectes i visibilitat multiavaluada d'objectes. L'existència d'una *visibilitat multiavaluada* $a \rightarrow B^*$ la podem veure com una *visibilitat monoavaluada sobre el multiobjecte* que engloba les diferents destinacions de l'enllaç en qüestió. És a dir, l'enllaç dirigit $a \rightarrow B^*$ el podem veure com un enllaç dirigit $a \rightarrow x$, on *x* és un multiobjecte de realitzacions del component *B* (és a dir, $x : B^*$).

Notació. Quan calgui remarcar la necessitat (o la possibilitat) de veure una visibilitat multiavaluada $a \rightarrow B^*$ com una visibilitat monoavaluada sobre un multiobjecte, expressarem la visibilitat com $a \rightarrow [B^*]$.



²Una alternativa igualment vàlida és exigir a la PRE l'existència de l'objecte que es pretén seleccionar. Llavors però cal afegir una operació *existeix(id)* que permeti consultar l'existència d'un determinat identificador dins del multiobjecte.

³Tal i com hem definit els multiobjectes l'*homogeneïtat* es dona sempre.

Dualitat unitat-col·lectiu en la notació (1). La notació $a \rightarrow [B^*]$ expressa la *dualitat unitat-col·lectiu* dels multiobjectes: per una banda a té visibilitat d'un *col·lectiu* de realitzacions de B (idea que expressa la *iteració* $*$); per una altra, a té visibilitat d'una *unitat* en forma d'un *multiobjecte* (idea que expressa l'*agrupador* $[\cdot]$).

Multiobjectes i visibilitat multiavaluada de components. L'existència d'una *visibilitat multiavaluada* $A \rightarrow B^*$ diu que existeix alguna realització $a : A$ que té visibilitat multiavaluada sobre B : $a \rightarrow B^*$. Per tant, si usem els *multiobjectes*, que existeix una realització $a : A$ tal que $a \rightarrow [B^*]$.



Notació. Quan calgui remarcar la necessitat (o la possibilitat) de veure una visibilitat multiavaluada $A \rightarrow B^*$ com una visibilitat monoavaluada sobre un multiobjecte, expressarem la visibilitat com $A \rightarrow [B^*]$.

Dualitat unitat-col·lectiu en la notació (2). La notació $A \rightarrow [B^*]$ expressa la possibilitat que cada realització de A tingui visibilitat sobre un *col·lectiu* de realitzacions de B (idea que expressa la *iteració* $*$); o bé (com a visió alternativa), que cada realització de A tingui visibilitat sobre una *unitat* o *multiobjecte* (idea que expressa l'*agrupador* $[\cdot]$).

2 Col·laboració i multiobjectes

2.1 Mecanisme homogeni de col·laboració

Homogeneïtat en les col·laboracions. Com veurem tot seguit, els multiobjectes permeten considerar un sol tipus de col·laboració: la *col·laboració d'enllaç*. \rightarrow A canvi, però, cal considerar dos tipus d'operacions diferents: les d'objecte i les de grup. Les *operacions d'objecte* es propaguen a tots els objectes que constitueixen el multiobjecte; les *operacions de grup* afecten el multiobjecte com a una unitat.

Dos tipus d'operacions. Anomenem *operacions de grup* les operacions que ofereix un multiobjecte; en la nostra presentació, `find()` i `add()`. Tota altra operació l'anomenarem *operació d'objecte*.

Col·laboració d'enllaç. L'ús dels multiobjectes permet expressar la *col·laboració d'enllaç* com l'enviament d'un missatge al *multiobjecte* corresponent; n'hi ha prou en assegurar que el multiobjecte interpreti "el missatge que rep com una operació d'objecte".

El multiobjecte com a "intèrpret" d'un missatge. El mecanisme intern del multiobjecte és l'encarregat de determinar si el missatge rebut es correspon a una operació d'objecte o de grup. En el cas de tractar-se d'una *operació d'objecte* és el mecanisme intern del multiobjecte qui es responsabilitza de fer arribar el missatge a tots els objectes que conté. I ho fa de manera transparent al dissenyador.

Excursió. (Aventatges del multiobjecte) La delegació de la gestió de la multiplicitat de l'enllaç en el multiobjecte té una conseqüència important: la independència del disseny respecte la implementació.

En el disseny, per expressar que un missatge ha d'arribar a tots els destinataris d'un enllaç dirigit (això és, per expressar una col·laboració d'enllaç a través d'una visibilitat multiavaluada) enviem el missatge al multiobjecte corresponent.

En implementació, si el llenguatge de programació no admet enviar un sol missatge a tots els objectes que són destinació d'un mateix enllaç (o simplement no admet els multiobjectes) n'hi haurà prou en implementar el mecanisme de replicació del missatge. El disseny es manté invariable. El que canvia és que una operació que el disseny considerava atòmica, en la implementació cal construir-la.

Col·laboració amb selecció. Amb l'ús dels multiobjectes, la *col·laboració amb selecció*[→] es converteix en dues col·laboracions: primer es fa la selecció (amb el `find()`, que és una *operació de grup*), i després es col·labora explícitament amb l'objecte seleccionat.

Exemple 44 (Col·laboració amb selecció) *Les etapes consten de dos indrets singulars: l'origen i la finalització de l'etapa. El disseny proposa una visibilitat $Etapa \rightarrow Indret^2$.*

El missatge `mostraAllotjamentInici()` rebut per una $e:Etapa$ es pot delegar al multiobjecte $x:Indret^2$, a través d'un missatge de selecció sobre el primer element del multiobjecte. El mecanisme concret és el següent:

1. *Col·laboració amb una operació de grup sobre el multiobjecte $x:Indret^2$ per fer la selecció:*^{4,5}

```
Indret b = x.find(origen)
```

2. *Col·laboració directa sobre l'objecte seleccionat:*

```
b.mostraAllotjament()
```

Col·laboració amb el grup. Amb l'ús dels multiobjectes la *col·laboració amb el grup* queda delimitada a les operacions ofertes pels multiobjectes, és a dir, les *operacions de grup*. D'aquesta manera la *col·laboració amb el grup* es converteix en una *col·laboració d'enllaç*, via una *operació de grup*, amb el multiobjecte que manté el grup.

⁵El `find()` tal com s'ha presentat implica una cerca per clau. Aquest exemple pressuposa una cerca per posició (`origen` és un símbol que expressa la primera posició), i per tant caldria ajustar l'especificació de l'operació `find()`.

⁵El resultat del `find()` és la creació d'un enllaç dirigit que com a destinació té l'objecte amb l'identificador donat. És a través d'aquest enllaç dirigit que després podrem enviar el missatge `mostraAllotjament()`.

Excursió. (UML) En UML les col·laboracions que afecten a tots els objectes del multiobjecte (és a dir, les col·laboracions d'enllaç segons una visibilitat multiavaluada) es marquen amb una indicació d'iteració (l'asterisc $*$) sobre la destinació de l'enllaç dirigit emprat per fer la col·laboració.

La manca d'aquesta marca d'iteració s'interpreta com que el missatge afecta al grup com a tal (i que per tant es tracta d'una col·laboració amb el grup; en la presentació donada aquí això només passa amb les operacions `find(id)` i `add(elem)`).

L'UML considera les operacions de grup com el cas no marcat; i marca les operacions d'objecte; és a dir, considera la col·laboració de grup el cas no marcat o habitual, i les col·laboracions d'enllaç com el cas marcat. En la presentació feta aquí les coses són a l'inrevés: considerem com a cas "normal" les col·laboracions d'enllaç (operacions d'objecte), i com a cas marcat les col·laboracions de grup (operacions de grup).

El mecanisme de col·laboració. En conclusió, el concepte de multiobjecte permet considerar la *col·laboració d'enllaç* com l'únic mecanisme de col·laboració. El missatge s'envia a través de l'enllaç, i es propaga a tots els objectes destinació de l'enllaç, a excepció del cas en què el missatge rebut es correspongui amb una operació de grup. En aquest cas, el destinatari del missatge és el propi multiobjecte com a unitat.

(15)

Donada una visibilitat $A \rightarrow [B^*]$:

- Les operacions d'objecte *traspassen* la "barrera" $[.]$ del multiobjecte
- Les operacions de grup *s'aturen* a la "barrera" $[.]$ del multiobjecte

2.2 Multiobjectes en una cadena de col·laboracions

Context de l'anàlisi. Sigui la cadena de visibilitats $A \rightarrow B^* \rightarrow C^*$. Sigui un objecte $a : A$ que envia un missatge $m()$ al multiobjecte $b : B^*$ corresponent (és a dir, $a : A$ usa l'enllaç dirigit $a \rightarrow [B^*]$ per enviar $m()$). Suposem que per exercir la responsabilitat $m()$ el component B requereix algun tipus de col·laboració, mitjançant el missatge $m1()$, amb el component C .

Objectiu de l'anàlisi. En el que segueix ens interessa estudiar les possibilitats de col·laboració del multiobjecte $b : B^*$, a través de la visibilitat $B \rightarrow C^*$, amb una (o més) realitzacions de C . I per fer-ho considerarem els possibles missatges $m1()$ emesos pel multiobjecte $b : B^*$ dins el context d'emissió $m()$.

La barrera del multiobjecte. Hem vist que l'ús dels multiobjectes permet considerar només les col·laboracions d'enllaç. En el cas que ens ocupa, però, si el context d'emissió $m()$ és una *operació de grup* aquesta no supera la *barrera* del multiobjecte, i per tant no és possible cap col·laboració amb C .

Traspàs de la barrera del multiobjecte. En el cas que el context d'emissió $m()$ sigui una *operació d'objecte*, el missatge $m()$ arriba a tots i cadascun dels objectes que componen el multiobjecte $b : B^*$. Això significa que la necessitat postulada de col·laboració amb el component C és una necessitat per cadascun dels objectes que rep el missatge $m()$. És a dir, la recepció, per part del multiobjecte $b : B^*$, d'un missatge $m()$ genera tants missatges $m1()$ com objectes contingui el multiobjecte.

Múltiples enllaços dirigits. El mecanisme intern dels multiobjectes és el responsable que el missatge $m()$ arribi a tots els objectes que engloba el multiobjecte. La necessitat de col·laboració postulada en el context de l'anàlisi d'aquest apartat significa que cadascun dels objectes $b_i : B$ destinació de l'enllaç dirigit $a \rightarrow B^*$ emeti el seu propi missatge $m1()$ a través de la visibilitat $B \rightarrow C^*$. Però això significa que cadascun dels objectes $b_i : B$ usará el seu propi enllaç dirigit $b_i \rightarrow C^*$.

Transparència del multiobjecte. Un missatge que correspongui a una *operació d'objecte* arriba a tots i cadascun dels objectes destinació de l'enllaç dirigit emprat; cadascun dels receptors d'aquest missatge el veu com un missatge propi i respon en conseqüència. El resultat és que l'ús del multiobjecte per simplificar les col·laboracions a través les visibilitats multiavaluades no afecta les cadenes col·laboratives.

Multiobjectes i visibilitat indirecta. La combinació de les visibilitats $A \rightarrow B \rightarrow C^*$ en forma de la visibilitat indirecta $A \xrightarrow{*} C^*$ es fa tenint en compte els enllaços dirigits presents en les visibilitats combinades. El mecanisme de traspàs de les barreres dels multiobjectes segueix exactament el mateix criteri. D'aquesta manera la col·laboració usant cadenes de visibilitat directa es comporta de manera consistent a l'ús de la visibilitat indirecta equivalent. Aquesta observació reforça les observacions fetes en l'apartat 3.3.4. *Visibilitat indirecta*, \rightarrow en el sentit que als efectes de possibilitats de col·laboració tota cadena de visibilitats directa és equivalent a la corresponent visibilitat indirecta.

Exemple 45 (Iteració de missatges per un multiobjecte (1))

Cada etapa conté diferents indrets singulars; cada indret té exactament un allotjament recomanat per l'organització de les caminades. El disseny proposat conté la següent cadena de visibilitats: $\text{Etapa} \rightarrow \text{Indret}^ \rightarrow \text{Allotjament}$.*

Suposem un missatge `mostraAllotjament()` que mostra els allotjaments recomanats per una etapa (un per cada indret singular de l'etapa). El disseny assigna aquesta responsabilitat a `Etapa`, qui al seu torn la delega en `Indret`.

El resultat és el següent. Quan una $e:\text{Etapa}$ rep el missatge `mostraAllotjament()`, el delega al multiobjecte corresponent a la visibilitat $\text{Etapa} \rightarrow \text{Indret}^$. Aquest missatge arriba a tots i cadascun dels indrets de l'etapa $e:\text{Etapa}$. Cadascun dels $i:\text{Indret}$ del multiobjecte delega, segons la visibilitat $\text{Indret} \rightarrow \text{Allotjament}$, el missatge `mostraAllotjament()` a l' $a_i:\text{Allotjament}$ corresponent.*

En conclusió, un missatge a un multiobjecte ha generat tants missatges com objectes contenia el multiobjecte.

Exemple 46 (Iteració de missatges per un multiobjecte (2))

En l'exemple 45, pàgina 96 \rightarrow la visibilitat de delegació des del multiobjecte era una visibilitat monoavaluada. Anem a considerar ara el cas d'una visibilitat multiavaluada: considerarem que un indret pot tenir múltiples allotjaments recomanats. Així tenim la següent cadena de visibilitats: $\text{Etapa} \rightarrow \text{Indret}^ \rightarrow \text{Allotjament}^*$.*

Anàlogament a l'exemple citat, el missatge `mostraAllotjament()` que l' $e:\text{Etapa}$ delega en el multiobjecte $a:\text{Indret}^$ arriba a cadascun del $i:\text{Indret}$ que componen el multiobjecte.*

Ara però, cada $i:\text{Indret}$ delega el missatge en el conjunt dels seus allotjaments. És a dir, cada $i:\text{Indret}$ envia un missatge `mostraAllotjament()` al multiobjecte pertinent segons la visibilitat $\text{Indret} \rightarrow \text{Allotjament}^$.*

El resultat és que una col·laboració d'enllaç amb un multiobjecte ha generat tantes col·laboracions d'enllaç com objectes contenia el multiobjecte. Però aquest és exactament el mateix cas que el de l'exemple 45: un missatge a un multiobjecte ha generat tants missatges com objectes contenia el multiobjecte. La diferència rau en què ara tenim més multiobjectes.

3 El multiobjecte com a artifici

El multiobjecte vist com un objecte. Tal com acabem de veure el multiobjecte es comporta, pel que fa a les col·laboracions, exactament igual que un objecte. El que diferencia un multiobjecte dels objectes "típics" és com respon als missatges rebuts.

Actuació d'un multiobjecte davant la recepció d'un missatge. En rebre un missatge un *multiobjecte* pot tenir dos comportaments diferents:

- Retransmetre'l (d'alguna manera) als objectes que conté. En aquest cas fa de pur transmissor; veiem el multiobjecte com un *col·lectiu*.
- Considerar el missatge com a propi. En aquest cas el multiobjecte actua com una *unitat*

El multiobjecte com a mecanisme de col·laboració. El multiobjecte és un mecanisme per expressar col·laboracions; és com un objecte ja que pot col·laborar mitjançant el mecanisme de transmissió dels missatges. Per tant és en el *model col·laboratiu*[→] on tenen la seva raó de ser.

4.1.El model col·laboratiu, pàgina 81

El multiobjecte no és la realització de cap component. Un multiobjecte es pot veure com un objecte perquè pot col·laborar per rebre i enviar missatges. Però no és un objecte "real" perquè no és la realització de cap component. Un multiobjecte és la múltiple realització d'un component; és a dir, un multiobjecte és un artífici que permet considerar molts objectes com un de sol allà on ens convingui.

Multiobjectes i model de components. Els *multiobjectes* no poden aparèixer en el model de components: no són un component, ni tampoc són la realització de cap component. Els multiobjectes són l'artífici emprat per poder expressar col·laboracions davant les visibilitats multiavaluades; en el *model de components*[→] el que apareix són aquestes *visibilitats multiavaluades*.

4.2.1.Expressivitat i contingut de MComp, pàgina 81

Multiobjectes i models del disseny.

- Els multiobjectes són un mecanisme de col·laboració
Han d'aparèixer en el *Model del comportament*
- Els multiobjectes no són la realització de cap component
NO han d'aparèixer en el *Model de components*

(16)

4 Incursió en la implementació

Multiobjectes no suportats. En la majoria dels llenguatges de programació imperatius cada variable només pot contenir un valor. Com a conseqüència els *multiobjectes*, com a mecanisme per mantenir múltiples destinacions sobre un mateix enllaç dirigit, són inviables.

Simulació dels multiobjectes. Malgrat tot, els llenguatges que no soporten els multiobjectes acostumen a incorporar mecanismes per a poder accedir a diferents valors a partir d'un mateix identificador.

Exemple 47 (Array) *En C el mecanisme per excel·lència és l'array: un mateix identificador permet mantenir diferents valors en posicions de memòria consecutives.*

Exemple 48 (Contenidors o col·leccions) *Llenguatges com el C++ o el Java ofereixen mecanismes per construir objectes que internament en continguin d'altres. En C++ són els contenidors (containers); en Java són les col·leccions (interfície *collection*).*

*Exemples concrets en Java són les interfícies (en terminologia Java) *Set*, *List* i *Map*; i les implementacions concretes d'aquestes com per exemple *ArrayList*, *LinkedList*, *HashMap* i *TreeSet*.*

Operacions sobre els multiobjectes simulats. Els mecanismes de simulació dels multiobjectes ofereixen operacions similars a les exposades aquí en la presentació dels multiobjectes.

Exemple 49 (Accés a multiobjectes simulats) *Als diferents valors d'un array del C s'hi accedeix posicionalment: tant en la recuperació (l'equivalent al *find()* dels multiobjecte) com en la inserció (l'equivalent a l'*add()* dels multiobjecte) cal indicar la posició de l'element afectat.*

*En Java a les col·leccions s'hi accedeix seqüencialment en la recuperació (mitjançant *iterators*); mentre que la inserció queda indeterminada: després d'una inserció el valor inserit es podrà recuperar de la col·lecció, però no sabem en quina posició. Les diferents subinterfícies i implementacions permeten redefinir aquest comportament.*

Necessitat de gestionar els missatges sobre multiobjectes. En enviar un missatge a un multiobjecte assumim que aquest té un mecanisme intern que fa arribar el missatge a tots els objectes destinació de l'enllaç dirigit emprat. En el cas d'usar un dels mecanismes d'implementació comentats per tal de simular un multiobjecte, aquest mecanisme intern és inexistent. Per tant cal que el programador es preocupi explícitament de fer arribar el missatge a cadascun dels seus destinataris; li cal *gestionar* els missatges sobre els multiobjectes.

Dues gestions possibles. Per a la gestió dels missatges hi ha dues possibilitats: l'*emissió gestionada* i la *recepció gestionada*.

Emissió gestionada. En l'*emissió gestionada* és responsabilitat de l'emissor del missatge assegurar que aquest arriba a tots els seus destinataris. El mecanisme més simple és "replicar" el missatge i enviar-lo individualment a tots i cadascun dels seus destinataris.

Exemple 50 (Emissió gestionada) *Sigui una visibilitat $A \rightarrow B^*$. El codi Java per enviar un missatge `m1` a través d'aquesta visibilitat, i sota el context d'emissió `m()`, sota la decisió d'emissió gestionada seria:*

```
class A
{
    List<B> elsB;    // Enllaç dirigit multiavaluat

    void m()
    {
        Iterator<B> i = elsB.iteraror(); // Retorna un iterador

        while(i.hasNext())                // Itera sobre elsB
            i.next().m1();
    }
}
```

Recepció gestionada. En el cas de la *recepció gestionada* el mecanisme de replicació del missatge per fer-lo arribar a tots els seus destinataris és transparent pel clients del multiobjecte (és a dir, per l'emissor del missatge sobre el multiobjecte). Per obtenir aquesta transparència, ja que la simulació del multiobjecte no ofereix el mecanisme de gestió dels missatges, caldrà accedir al multiobjecte a través d'un intermediari que faci la gestió dels missatges. És a dir, l'emissor *delega* en un gestor, qui al seu torn es preocuparà de fer una emissió gestionada del missatge que se li ha delegat.

Exemple 51 (Recepció gestionada) *L'exemple 50, en el cas de tenir recepció gestionada, quedaria:*

```
class A
{
    Gestor g = new Gestor();

    void m()
    {
        g.m(); // Delegació sobre el gestor
    }
}

class Gestor
{
    List<B> elsB;    // Enllaç dirigit multiavaluat

    void m()
    {
        Iterator<B> i = elsB.iteraror(); // Retorna
        // un iterador
    }
}
```

```
while (i.hasNext())           // Itera sobre els B
i.next().m1();
}
```

Implementació d'un multiobjecte. En cas d'haver d'implementar amb un llenguatge que no disposi de multiobjectes, caldrà simular-los amb els mecanismes que ofereixi el llenguatge, i (segurament) caldrà preocupar-se de gestionar els missatges sobre els multiobjectes. Sigui com sigui el disseny continua sent vàlid: si la implementació no ofereix allò que necessitem, ho afegim, però no ens cal modificar res.

5 MOMO: Un modisme pels multiobjectes

Ús dels modismes. Un *modisme* és un conjunt de decisions i principis de disseny que es fonamenten en l'ús d'un llenguatge o tecnologia determinada. Un modisme no és pròpiament una decisió d'implementació, però hi està molt propera: el que el diferencia d'una decisió d'implementació és la seva generalitat, en el sentit que un modisme és aplicable a molts llenguatges o tecnologies.

Multiobjectes i modismes. Són tants els llenguatges de programació que no admeten el mecanisme dels multiobjectes que té sentit presentar un *modisme* sobre ells. Si bé en el disseny el mecanisme dels *multiobjectes* és del tot transparent (de fet només és una ficció que ens permet treballar còmodament amb la dualitat unitat-col·lectiu), gran part dels llenguatges de programació n'exigeixen una implementació explícita (i per tant es perd la transparència). Aquesta exigència generalitzada és el que justifica la presentació de *modismes* per tractar l'especificitat dels multiobjectes.

Principi



MOMO: Modisme de l'Objecte MultiObjecte. En el disseny cal considerar un objecte per cada multiobjecte que hi aparegui

El modisme MOMO. Els multiobjectes són una ficció útil per expressar les col·laboracions a través d'enllaços dirigits multiavaluats. L'ús dels multiobjectes pressuposa un mecanisme determinat, que pot ser intern al multiobjecte, o pot ser gestionat des del disseny. El modisme MOMO diu que cal considerar els multiobjectes com a objectes, i per tant el disseny és responsable del seu cicle de vida.

Objecte sense component. Des d'una perspectiva pura del disseny, no existeix cap component del qual el multiobjecte en pugui ser una realització,[→] i per això podem afirmar que un multiobjecte no és pas un objecte. El que proposa el modisme MOMO és considerar el multiobjecte com un objecte que és realització d'un component inexistent o fictici.

MOMO i les necessitats de creació. L'objecte multiobjecte proposat per MOMO és el receptor de les col·laboracions de grup. Per tant, respecte a aquestes, es comporta com un objecte qualsevol. D'aquí l'exigència que calgui crear l'objecte multiobjecte abans que pugui respondre a cap petició.

Responsabilitats sobre els multiobjectes. L'ús del modisme MOMO fa que sigui responsabilitat del disseny decidir qui crea els multiobjectes i quan. Per contra, és responsabilitat de la implementació assegurar que el mecanisme de captura dels missatges de grup i el de propagació dels missatges d'enllaç es comportin tal com s'han explicat en presentar l'artifici dels multiobjectes. →

Ús del modisme MOMO. La diferència entre usar o no el modisme MOMO rau en la necessitat d'assignar o no la responsabilitat de creació del multiobjecte. Si usem el modisme estem baixant força el nivell de disseny, i pot ser que estiguem fent una feina innecessària (perquè el llenguatge de programació que usarem finalment sí admet els multiobjectes). Si no usem el modisme, si al final anem a parar a un llenguatge de programació que no admet els multiobjectes caldrà analitzar la creació d'aquests durant la implementació, la qual cosa ens obliga a treballar amb la globalitat del sistema (i justament una de les gràcies del disseny és aconseguir que les decisions d'implementació siguin al màxim de locals possible).

Recomanació sobre l'ús de MOMO. El risc en cas de no emprar el modisme MOMO és gran: la probabilitat d'usar un llenguatge de programació que no admeti els multiobjectes és gran; alhora que el cost d'analitzar la creació dels multiobjectes en la implementació és elevat. L'ús innecessari del modisme no és un greuge massa elevat: la innecessarietat és poc provable ja que la majoria de llenguatges de programació no admeten els multiobjectes; per altra banda, la intoxicació en el disseny és mínima i afitada, ja que es redueix a la introducció d'una nova decisió d'assignació de responsabilitat. D'aquí la recomanació d'usar sempre el modisme MOMO.

El disseny haurà de resoldre l'assignació de responsabilitat de creació dels multiobjectes⁶

(9) ? !

⁶És a dir, exigim l'aplicació de MOMO, que d'aquesta manera deixa de ser un principi i es converteix en una *regla*.

Principis i definicions del capítol

Principis

MOMO: Modisme
de l'Objecte
MultiObjecte, 100

Definicions

Multiobjecte, 90

Part II

Comencem a caminar

Part II: Taula de continguts

6	Primers principis de disseny	107
1	L'expertesa és un grau	110
2	El paper dels controladors	112
3	Necessitat de comunicació entre els ES	113
4	Responsabilitats dels controladors	115
	Principis	116
7	<i>GLS</i> Cas d'ús <i>ferInscripció</i>	117
1	Comencem a dissenyar	119
2	Contracte de <i>novaInscripcio(c:Caminada)</i>	119
3	POST: Crear una nova <i>i:Inscripció</i>	120
4	Anàlisi d'alternatives	123
8	Anàlisi d'alternatives	125
1	Necessitat dels principis avaluatius	127
2	Cohesió	127
3	Acoblament	128
4	Anàlisi de l' <i>Encarrilament</i>	128
	Principis	130
	Definicions	130
9	Una patata que crema	131
1	<i>GLS</i> Tasca pendent	135
2	El problema de la patata calenta	135
3	Ús, emmagatzematge i creació	140
4	Repositoris	147
	Definicions	150

10 Principis de creació	151
1 Principi <i>Creador (GRASP)</i>	158
2 Principi <i>Expert delegat</i>	159
3 Principi <i>Creador orb</i>	161
4 Emmagatzematge	164
5 Principi del <i>Creador canònic</i>	173
6 Emmagatzematge primari	176
7 Principi del <i>Creador generalitzat</i>	182
8 Migració i compartició de realitzacions	183
9 Ús dels diferents principis de creació	187
10 La patata calenta i els principis de creació	189
11 Creació i recuperació de la informació	191
Principis	197
Definicions	197
11 ^{GLS}L'ES novaInscripció	199
1 La <i>i:Inscripció</i> com a patata calenta	202
2 POST: Enllaçar <i>i i c</i>	215
12 Arguments implícits	217
1 Realització activa	220
2 Manteniment consistent dels contractes	224
3 Parèntesi sobre l'eficiència	227
Principis	229
Definicions	229
13 ^{GLS}POST: La <i>i:Inscripció</i> es manté activa	231
1 Activació de la <i>i:Inscripció</i>	235
2 Activació a <i>M1</i>	235
3 Activació a <i>M2</i>	239
4 Anàlisi de l'eficiència	242
5 Conclusions i decisions de disseny	245
6 Els models en la situació actual	247

Capítol 6

Primers principis de disseny

1	L'expertesa és un grau	110
1.1	Un principi general	110
1.2	Lectura dels principis d'assignació	111
2	El paper dels controladors	112
2.1	Primeres responsabilitats	112
2.2	Capes de <i>Presentació</i> i de <i>Domini</i>	113
3	Necessitat de comunicació entre els ES	113
3.1	El controlador com a mecanisme de comunicació	113
3.2	Controlador de CU artificial	114
4	Responsabilitats dels controladors	115
	Principis	116

Contingut detallat del capítol 6

1 L'expertesa és un grau	110
1.1 Un principi general	110
Principis d'assignació de responsabilitat	110
Principis <i>GRASP</i>	110
<i>GRASP</i> (General Responsibility Assignment Software Principles)	110
Principi bàsic universal	110
Expert (<i>GRASP</i>)	110
Experts parcials	110
1.2 Lectura dels principis d'assignació	111
Recordatori sobre l'assignació de responsabilitats	111
Una expressió, dues lectures	111
El principi <i>Expert</i> en termes d'objectes	111
<i>Expert</i> : objecte i component	111
<i>Experts parcials</i> : segona part	111
<i>Expert parcial</i> : objecte i component	111
Col·laboració entre experts parcials	111
2 El paper dels controladors	112
2.1 Primeres responsabilitats	112
Descomposició inicial	112
2.1.1 Captura de l'esdeveniment de sistema	112
Petició externa	112
Captura de l'ES	112
2.1.2 La captura de l'ES segons <i>Expert</i>	112
Una responsabilitat compartida	112
2.1.3 Assignar la captura de l'ES	112
Expertesa en tot	112
Controlador (<i>GRASP</i>)	112
Controlador: objecte i component	112
2.2 Capes de <i>Presentació</i> i de <i>Domini</i>	113
Emissor i receptor dels ES	113
<i>Experts</i> en el domini	113
Desacoblament entre <i>Presentació</i> i <i>Domini</i>	113
3 Necessitat de comunicació entre els ES	113
Col·laboració entre els ES	113
Asincronia dels ES	113
3.1 El controlador com a mecanisme de comunicació	113
Un punt de contacte	113
Un controlador per a tots	113
Mecanisme de comunicació	113
Controlador de CU	114
Controlador d'un ES i controlador d'un CU	114
No localitat de la comunicació	114

Propietats comunicables	114
3.2 Controlador de CU artificial	114
Un possible recurs	114
Violació del principi de l' <i>Espill</i>	114
Excursió. (Acoblament i cohesió)	115
4 Responsabilitats dels controladors	115
Encarilament	115
El <i>controlador</i> com a recepcionista	115
Excursió. (Fabricació pura i cohesió)	115
Principis	116

1 L'expertesa és un grau

1.1 Un principi general

2.3.Les tasques del dissenyador, pàgina 24

Principis d'assignació de responsabilitat. La *descomposició*^{1,→} ens detecta una responsabilitat R . El problema en l'assignació és determinar quin component s'ha de fer responsable de R .

Principis GRASP. Els principis GRASP són un dels conjunts de principis d'assignació de responsabilitats més coneguts; i formen part del vocabulari de tot dissenyador.

GRASP (General Responsibility Assignment Software Principles)

Principis bàsics de disseny per a l'assignació de responsabilitats al software, introduïts per [Larman]. Els principis GRASP principals són:

1. *Expert*. Principi bàsic d'assignació.
2. *Controlador*. Particularització del principi bàsic al cas dels esdeveniments de sistema.
3. *Creador*. Particularització del principi bàsic en el cas que calgui crear noves realitzacions.
4. *Alta Cohesió*. Principi avaluatiu que permet contrastar alternatives de disseny.
5. *Baix Acoblament*. Principi avaluatiu que permet contrastar alternatives de disseny.
6. *Fabricació Pura*. Principi que posa les condicions sota les quals es permet "violat" el principi de l'*Espill*

Principi bàsic universal. Quin criteri bàsic i universal cal seguir per assignar responsabilitats? La resposta la dona el principi *Expert*.

Principi



Expert (GRASP). La responsabilitat R l'assignem a aquell component C que té tota la informació necessària per poder realitzar la responsabilitat demanada. Diem que C és l'*expert* de R .

Experts parcials. Sovint el que tenim són *experts parcials* que han de col·laborar entre ells.²

¹A l'apartat 2.3.Les tasques del dissenyador, pàgina 24, s'han introduït les tasques d'assignació, descomposició i recomposició col·laborativa.

²Acabem de definir *expert* com un component; i ara parlem de col·laboració, la qual cosa exigeix parlar d'objectes i no pas de components. Aquesta aparent contradicció es resol en el següent apartat.

1.2 Lectura dels principis d'assignació

Recordatori sobre l'assignació de responsabilitats. Si bé les responsabilitats s'assignen als components, són els objectes que en són realització qui les realitzen. D'aquí que podem parlar de dos nivells d'assignació d'una responsabilitat: l'assignació a un component i l'assignació a un objecte. L'assignació a un component és la tasca pròpiament dita d'*assignació de responsabilitat*; l'assignació a un objecte és la tasca que anomenem *recomposició col·laborativa*.^{3,→}

3.1.2.Responsabilitats, pàgina 37

Una expressió, dues lectures. Els principis d'assignació de responsabilitats, com el principi *Expert*, s'expressen en termes de *components*. Cal però tenir present que en la seva aplicació hi ha involucrada l'exercici de la responsabilitat que s'està assignant; és a dir, la *recomposició col·laborativa*. Això exigeix llegir també els principis en termes d'*objectes*.

El principi *Expert* en termes d'objectes. Llegit en termes d'objectes, el principi *Expert* diu que l'exercici de la responsabilitat R l'assignem a aquell objecte $c : C$ que té tota la informació necessària per poder realitzar la responsabilitat demanada. Diem que $c : C$ és l'*expert* en R .

Expert: objecte i component. El que diu el principi *Expert* és que una responsabilitat s'ha d'assignar a aquell que en sigui *expert* en la responsabilitat. Però des del punt de vista de l'*assignació de responsabilitats* l'*expert* és un component; mentre que des del punt de vista de la *recomposició col·laborativa* l'*expert* és un objecte.

Experts parcials: segona part. Més amunt hem comentat com la presència d'*experts parcials* requereix la col·laboració entre ells. Ara tenim tots els elements per poder entendre aquesta afirmació. Ho veiem tot seguit.

Expert parcial: objecte i component. Un *expert parcial* sobre una responsabilitat R és aquell qui té part de la informació necessària per exercir la responsabilitat R , però no la té tota. Des del punt de vista de l'*assignació de responsabilitats* l'*expert parcial* és un component; mentre que des del punt de vista de la *recomposició col·laborativa* l'*expert parcial* és un objecte.

Col·laboració entre experts parcials. L'existència d'*experts parcials* (juntament amb l'absència d'un *expert* total) exigeix que en la *recomposició col·laborativa* aquests *experts parcials* col·laborin entre ells. És a dir, són els objectes *experts parcials* els qui col·laboren; i aquests han de ser realitzacions dels components *experts parcials*.

³A l'apartat 3.1.2.Responsabilitats, pàgina 37, es fa el plantejament en detall d'aquestes afirmacions.

2 El paper dels controladors

2.1 Primeres responsabilitats

Descomposició inicial. La descomposició inicial ens ve donada per l'especificació en forma d'esdeveniments de sistema. Per tant les primeres responsabilitats per assignar i resoldre són les corresponents als diferents ES.

2.1.1 Captura de l'esdeveniment de sistema

Petició externa. Quan un agent extern fa una petició al sistema, a través d'un ES, està demanant al sistema que resolgui les responsabilitats expressades en el contracte de l'ES.

Captura de l'ES. Abans, però, de dissenyar, en termes d'assignació i recomposició, com resoldre les responsabilitats que se'ns demanen externament, cal assignar la responsabilitat de respondre a la petició externa (que ens arriba en forma d'ES). Cal assignar la responsabilitat de *capturar* cadascun dels ES.

2.1.2 La captura de l'ES segons *Expert*

Una responsabilitat compartida. El principi *Expert* diu que el responsable és aquell qui tingui el coneixement necessari. Òbviament cada ES té un contracte diferent, i per tant unes necessitats d'informació diferents; però tots ells són peticions externes al sistema: "Et demano que facis X". El que canvia és que demanem exactament (la X); però la responsabilitat per assignar és la mateixa: capturar la petició externa per tal de poder-la respondre.

2.1.3 Assignar la captura de l'ES

Expertesa en tot. Qui és l'*expert* en la captura d'un ES? Ha de ser algun component que pugui respondre a qualsevol petició X feta des de l'exterior del sistema, sigui quina sigui aquesta. Per tant ha de ser algun component que sigui *expert* en tot el sistema; és a dir, ha de ser algun component que representi la totalitat del sistema o del subsistema considerat.

Principi



Controlador (GRASP). La responsabilitat de respondre a la petició d'un ES l'assignem a un component C que representi o modelitzi la totalitat del sistema o del subsistema considerat. El component C rep el nom de *controlador* d'aquest ES.

Controlador: objecte i component. El *controlador* d'un ES és el component a qui el principi *Controlador* dona la responsabilitat de capturar aquest ES. Des del punt de vista de l'*assignació de responsabilitats* el *controlador* és un component; mentre que des del punt de vista de la *recomposició col·laborativa* el *controlador* és un objecte.

2.2 Capes de *Presentació* i de *Domini*

Emissor i receptor dels ES. Recordem que els ES són les peticions fetes al sistema, un cop superada la capa d'interacció amb l'usuari (o capa de *Presentació*). Per tant els ES els podem veure com les peticions que la capa de *Presentació* fa a la capa del *Domini*, que és el que fins ara hem estat anomenant *Sistema*.⁴ La *Presentació* emet els ES; el *Domini* els rep.

Experts en el domini. Els *controladors* esdevenen així els experts en el *domini* des del punt de vista de la *Presentació*: són els components que fan de pont entre la capa de *Presentació* i la capa de *Domini*.

Desacoblament entre *Presentació* i *Domini*. El principi *Controlador* diu quins components poden fer aquesta tasca de pont. L'existència dels *controladors* desacobla el *domini* i la *presentació*: per tal que la *presentació* pugui fer totes les peticions necessàries al *domini*, només ha de tenir visibilitat sobre els *controladors*.

3 Necessitat de comunicació entre els ES

Col·laboració entre els ES. Sovint els diferents ES necessaris per realitzar un mateix CU han de comunicar-se entre ells. Per exemple deixant com a POST l'estat que apareix a la PRE del següent ES de la seqüència.⁵

Asincronia dels ES. Els diferents ES que apareixen en un mateix CU poden generar-se en moments molt diferents; fins i tot en sessions diferents. De fet hem de veure els ES com a peticions asíncrones sobre el sistema. Aquesta asincronia dificulta la comunicació.

3.1 El controlador com a mecanisme de comunicació

Quin és el mecanisme de comunicació entre els diferents ES que participen conjuntament en la realització d'un mateix CU?

(10) ?

Un punt de contacte. Sabem que tots els ES tenen en comú que són capturats per un controlador. Podem usar aquesta informació per resoldre els problemes comunicatius entre els ES d'un mateix CU?

Un controlador per a tots. La idea es basa en assegurar que tots els ES que tenen necessitat de comunicar-se comparteixin el controlador. D'aquesta manera la comunicació es pot fer a través del controlador.

⁴Estem aplicant una arquitectura a dues capes. Però el dit aquí es manté vàlid en arquitectures n-capes.

⁵Recordem que en l'especificació els CU s'expressen en termes de seqüència d'ES.

Mecanisme de comunicació. La realització de la responsabilitat de capturar un ES exigeix un *objecte* controlador k . Aquest objecte pot tenir diferents propietats, que es mantenen en forma d'atributs o de visibilitats. Si un ES modifica aquestes propietats, i d'alguna manera podem assegurar que un segon ES serà capturat pel mateix objecte controlador k , llavors aquest segon ES podrà accedir, per consultar o per modificar, al valor d'aquestes propietats.

Principi



Controlador de CU. Tots els ES d'un mateix CU són capturats pel mateix *objecte* controlador.

Controlador d'un ES i controlador d'un CU. Aquest principi ens permet parlar de *controlador del CU*, i no del *controlador de l'ES*.

No localitat de la comunicació. L'exigència que dos ES comparteixin el mateix objecte controlador és per permetre la comunicació entre ells. Per exemple, la propietat que modifica un ES és usada per l'altre ES. Per tant les propietats que es poden comunicar han de tenir un àmbit de validesa que superi el de la realització de cadascun dels ES; és a dir, no poden ser propietats amb àmbit local sobre l'ES.

Propietats comunicables. De tot plegat en resulta que les propietats que volgum comunicar entre diferents ES han de ser atributs o visibilitats d'atribut de l'objecte controlador.

Definició. Propietat atributiva

Anomenem propietat atributiva a aquella informació que es manté amb un atribut o amb una visibilitat d'atribut

(17)

Comunicació entre ES.

Les propietats comunicables entre els ES són les *propietats atributives* de l'objecte controlador comú als ES que s'estan comunicant

3.2 Controlador de CU artificial

Un possible recurs. Si l'aplicació del principi del *Controlador de CU* porta problemes (no tenim cap component que representi tot el sistema o un subsistema pertinent, o potser si usem algun dels que tenim violem d'altres principis de disseny) sempre podem crear un component específic amb l'única finalitat de fer de controlador del CU.

Violació del principi de l'Espill. La decisió d'introduir un component específic, artificial, per fer de controlador del CU és una violació del principi de l'Espill: introduïm un component que no es correspon a cap element de la "realitat" modelitzada.

Excursió. (Acoblament i cohesió) El principi *Controlador* diu quins components poden fer de controlador. L'elecció d'un dels components possibles com a controlador ha de tenir en compte, però, d'altres principis de disseny com els principis d'*Alta cohesió* i *Baix acoblament*.[→]

El principi de la *Fabricació pura* estableix en quins casos pot ser útil violar, localment, el principi de l'Espill.

4 Responsabilitats dels controladors

Encarrilament. La tasca bàsica d'un controlador és la delegació de les responsabilitats que els ES li demanen.

Principi



El controlador com a recepcionista. Aquest principi ens diu que els controladors no fan res per si mateixos; són com els porters a les entrades dels edificis oficials que ens encarrilen cap a les dependències correctes. Ens reben, i ens escolten; però les solucions no ens les donen pas ells.

Excursió. (Fabricació pura i cohesió) El principi del *Controlador de CU* assegura una bona cohesió del controlador. L'ús de la *Fabricació pura* per decidir el controlador permet mantenir la cohesió en els casos que tinguem més d'un CU. El principi de l'*Encarrilament* augmenta la cohesió del controlador: només fa una tasca, la de respondre els esdeveniments de sistema; però les tasques concretes demanades per aquests, les fan d'altres components.

Principis i definicions del capítol

Principis

Controlador (*GRASP*), 112
Controlador de CU, 114

Encarrilament, 115
Expert (*GRASP*), 110

Capítol 7

GLS Cas d'ús *ferInscripció*

1	Comencem a dissenyar	119
1.1	Mètode de construcció de MComp	119
1.2	Itinerari del disseny	119
1.3	Convencions usades	119
2	Contracte de <i>novaInscripcio (c:Caminada)</i>	119
3	POST: Crear una nova i:Inscripció	120
3.1	Controlador	120
3.2	Components detectats	121
3.3	Expert	122
4	Anàlisi d'alternatives	123

Contingut detallat del capítol 7

1 Comencem a dissenyar	119
1.1 Mètode de construcció de MComp	119
1.2 Itinerari del disseny	119
Selecció d'un CU	119
Selecció d'un ES	119
Tasques pendents	119
1.3 Convencions usades	119
Raonaments generals i raonaments particulars	119
Homogeneïtat dels capítols	119
Dispersió del desenvolupament	119
La marca ^{GLS}	119
2 Contracte de novaInscripcio (c:Caminada)	119
Postcondicions	120
Excursió. (Incorporació tardana)	120
3 POST: Crear una nova i:Inscripció	120
3.1 Controlador	120
Responsabilitat a resoldre	120
Selecció del controlador dins MComp	120
Introducció d'un nou component a MComp	120
Com sortir de l'atzucac	120
Com modifiquem MC	120
Excursió. (Singleton)	121
Excursió. (Dificultats de modificar MC)	121
Després de la captura	121
3.2 Components detectats	121
Components obtinguts per <i>síntesi</i>	121
Components induïts pel contracte	121
Components induïts per la responsabilitat	121
Components actuals	122
3.3 Expert	122
Aplicació de <i>Expert</i>	122
Informació necessària per a la <i>Inscripció</i>	122
Un possible <i>expert</i> per la creació de <i>i</i>	122
Un expert alternatiu per a la creació de <i>i</i>	122
4 Anàlisi d'alternatives	123
Múltiples candidats	123
Principis avaluatius	123

1 Comencem a dissenyar

1.1 Mètode de construcció de MComp

Usarem el mètode de construcció sintètica del model de components.

1.2 Itinerari del disseny

Selecció d'un CU. Començarem per dissenyar el CU *ferInscripció*.

Selecció d'un ES. *novaInscripcio(c:Caminada)* és el primer ES que considerarem.

Tasques pendents. En capítols posteriors afrontarem els altres ES i els altres CU.

1.3 Convencions usades

Raonaments generals i raonaments particulars. Tindrem especial cura en distingir entre els raonaments que afecten a l'exemple concret de *Gastem la sola*, d'aquells raonaments genèrics.

Homogeneïtat dels capítols. Quan durant el disseny d'un ES de *Gastem la sola* calgui introduir nous conceptes o principis ho farem en capítols independents. D'aquesta manera els capítols seran *homogenis*, en el sentit que o bé tractaran una problemàtica general, o bé tractaran una problemàtica concreta (la de *Gastem la sola*).

Dispersió del desenvolupament. El criteri de la homogeneïtat dels capítols, així com el desig de no introduir més conceptes o principis dels necessaris en cada moment, fan que el desenvolupament d'un mateix ES pugui ocupar diferents capítols no consecutius.

La marca ^{GLS}. Tot capítol, com l'actual, que tracti alguna problemàtica concreta de *Gastem la sola* portarà el prefix ^{GLS} en el seu títol.

2 Contracte de *novaInscripcio(c:Caminada)*

Contracte *novaInscripcio(c:Caminada)*

Paràmetres

- *c*: Caminada a la que es vol fer la inscripció

PRE

1. No hi ha cap *Inscripció* activa

POST

1. Hi ha una nova realització *i*:*Inscripció*
2. Hi ha un enllaç entre *i* i *c*, corresponent a l'associació relativa *a*
3. La inscripció *i* passa a estar activa

Postcondicions Expressen *condicions* que s'han de complir, no pas *accions* que s'han realitzat. Per això usem expressions com “Hi ha” i no pas “Es crea” o “S'ha creat”.

Excursió. (Incorporació tardana) El principi de la *Incorporació tardana*[→] ens obligarà a canviar aquest contracte per tal de posposar la creació de l'enllaç entre *c* i *i* a l'ES *fiInscripcio()*.

De totes maneres, encara que apliquéssim ja de bones a primeres el principi de la *Incorporació tardana* hauríem de tenir present en les decisions de disseny la necessitat d'enllaçar *c* amb *i* en algun moment.

3 POST: Crear una nova *i*:*Inscripció*

? (11)

A qui assignem la responsabilitat de respondre a l'esdeveniment de sistema *novaInscripcio(c:Caminada)*?

3.1 Controlador

Responsabilitat a resoldre. Estem davant d'un ES. Per tant cal usar el principi *Controlador*.

Selecció del controlador dins MComp. El MComp construït fins ara és buit (acabem de començar i encara no hem fet res).

Introducció d'un nou component a MComp. Accedim al model conceptual (MC) a veure si hi ha algun concepte que representi tot el domini. En cas de trobar-lo, introduïm a MComp el component que és el correlat software d'aquest concepte. Problema: al nostre MC no hi ha cap concepte que representi tot el domini.


Com sortir de l'atzucac. Per sortir de l'atzucac on ens trobem podem:

- Afegir al MC un concepte que representi tot el domini
- Introduir un component que tingui justament aquesta funcionalitat (*Controlador de CU artificial*)

Com modifiquem MC. Si ens decantem per modificar el MC podem afegir un concepte GLS (*Gastem la sola*) que representa tot el domini. S'ha d'observar, però, que aquest concepte té una sola realització: el client que ens demana que construïm el sistema.

Excursió. (Singleton) A vegades en els MC apareixen conceptes dels quals sabem que només hi ha una realització. Per tant, el component que modelitza aquest concepte només pot tenir una realització. A nivell d'implementació, la classe que modelitza el component en qüestió només podrà tenir un objecte associat. Aquest problema es coneix amb el nom de *Singleton* i hi ha diversos mecanismes, en qualsevol de les etapes de desenvolupament, per tractar-lo. A nivell del MC o del MComp, en notació UML, s'indica l'existència d'un singleton amb un 1 a la part superior dreta interior de les caixes que representen els conceptes o els components.

Decidim no modificar el MC i introduir el concepte `GLS` (que violarà *Espill*). Assignem el paper de *controlador* al component `GLS`.

(12) 

Excursió. (Dificultats de modificar MC) La decisió d'afegir a MC un concepte que representi tot el domini no està lliure de dificultats. En concret cal analitzar quines associacions manté el nou concepte amb els altres conceptes. Per això aquí hem preferit no modificar *MC*, i violar *Espill*.

Després de la captura. Un cop decidit el controlador passem a analitzar les POST del contracte.

Qui és el responsable de crear la nova realització de `Inscripció`?

(13) ?

3.2 Components detectats

Components obtinguts per síntesi. Recordem que estem construint MComp pel mètode de *síntesi*.[→] De moment només tenim un component: el controlador `GLS`. Ara bé, per assignar les responsabilitats del CU que estem dissenyant, no n'hi ha prou en considerar els components ja detectats: quan convingui caldrà introduir nous components que siguin correlats d'algun concepte de MC.

4.5.Construcció del MComp, pàgina 85

Components induïts pel contracte. En l'especificació, apareix com a argument de l'ES *novaInscripcio* (`c:Caminada`) una realització de *Caminada*. Això exigeix que en el disseny l'ES tingui com a argument una realització del component *Caminada*. Es tracta d'un component que de moment no havíem considerat, però que el contracte ens exigeix.¹

¹El principi de les *Connexions* (pàgina 345)17.2.3., pàgina 345 ens dirà que pròpiament l'argument de l'ES és un *identificador* de *Caminada*, i no pas una realització de *Caminada* per sí mateixa. Malgrat tot, això no invalida l'argument que el contracte de l'ES exigeix la presència a MComp d'un component *Caminada*.

Components induïts per la responsabilitat. La responsabilitat bàsica d'aquest ES és la incorporació al sistema d'una nova *i:Inscripció*. En termes del disseny això significa la incorporació d'un nou objecte *i*, realització d'un component *Inscripció*.²

Components actuals. En el moment present, per tant, MComp té tres components: GLS, Caminada i *Inscripció*. El primer té una responsabilitat assignada (capturar l'ES: n'és el *controlador*); els altres dos no tenen cap responsabilitat assignada, de moment.

3.3 Expert

Aplicació de *Expert*. Aquest principi ens diu que el responsable és aquell qui té tota la informació necessària. Per tant, abans de res hem de respondre la pregunta:

? (14)

Què necessitem per crear una realització d'*Inscripció*?

15.Identificadors, pàgina 277

Informació necessària per a la *Inscripció*. L'identificador es crea internament^{3,→} i per tant ens en podem oblidar. Sabem però que cada *Inscripció* és per una *Caminada* concreta. Per tant sembla lògic suposar que ens cal conèixer la *c:Caminada* per poder crear la *i:Inscripció*.

Un possible *expert* per la creació de *i*. Qui coneix la *c:Caminada*? Doncs el controlador, ja que la rep com un argument de l'ES. Per tant el controlador és un bon candidat a qui delegar la responsabilitat de crear una nova *inscripció*.

Un expert alternatiu per a la creació de *i*. De totes maneres no és l'única possibilitat. La pròpia *c:Caminada* també és un *expert* per a la creació d'una nova *Inscripció*; i com que el controlador coneix la *c:Caminada* pot delegar-li la tasca de crear la *i:Inscripció*.

Responsable	Principi	Observacions
GLS	<i>Expert</i>	La <i>c:Caminada</i> és coneguda com a argument de l'ES
<i>c:Caminada</i>	<i>Expert</i>	La <i>c:Caminada</i> és l'únic que necessitem

Figura 7.1: Creadors d'*i:Inscripció* segons *Expert*

²De fet el component *Inscripció* també el podem veure com un component induït pel contracte. La POST de l'ES *novaInscripcio(c:Caminada)* exigeix l'existència d'un nou objecte *i:Inscripció*.

³Més endavant, apartat 15.Identificadors, pàgina 277, veurem com.

4 Anàlisi d'alternatives

Múltiples candidats. L'aplicació del principi *Expert* sobre la responsabilitat de crear una nova *i*: *Inscripció* ens ha proporcionat dos candidats possibles a qui assignar aquesta responsabilitat: el controlador GLS i la Caminada.

Com dedidim, entre els diferents candidats, quin és el més idoni?

(15) ?

Principis avaluatius. Una possible solució l'ofereix l'aplicació dels *principis avaluatius*, que presentem en el següent capítol.

Capítol 8

Anàlisi d'alternatives

1	Necessitat dels principis avaluatius	127
2	Cohesió	127
3	Acoblament	128
4	Anàlisi de l'Encarrilament	128
	Principis	130
	Definicions	130

Contingut detallat del capítol 8

1 Necessitat dels principis avaluatius	127
Principis, no reglesma	127
<i>GLS</i> Diferents alternatives	127
Ús dels principis avaluatius	127
2 Cohesió	127
Aproximació quantitatives a la cohesió des dels components	127
Aproximació quantitatives a la cohesió des dels objectes	127
Problemes d'una cohesió baixa	127
Alta cohesió (<i>GRASP</i>)	127
3 Acoblament	128
L'acoblament és coneixement	128
Aproximació quantitativa a l'acoblament	128
Problemes de l'alt acoblament	128
Excursió. (Acoblament i implementació)	128
Baix Acoblament (<i>GRASP</i>)	128
4 Anàlisi de l'Encarilament	128
Encarilament i Alta Cohesió	128
Encarilament i Baix Acoblament	128
Principis	130
Definicions	130

1 Necessitat dels principis avaluatius

Principis, no reglesma. Com hem vist els principis representen bons consells que és bo seguir, però no són pas una regla que necessàriament s'hagi de seguir.

GLS **Diferents alternatives.** En el capítol 7. *GLS* Cas d'ús *ferInscripció*,[→] en aplicar el principi *Expert* per resoldre la primera POST de l'ES *novaInscripció* (*c:Caminada*), s'han obtingut dues recomanacions diferents: tant Caminada com GLS es consideren bons candidats per a crear la Inscripció. Quina és la bona recomanació?

7. *GLS* Cas d'ús *ferInscripció*, pàgina 117

Ús dels principis avaluatius. Davant de diverses alternatives proposades bé per un mateix principi, bé per principis diferents, cal usar d'altres principis per decidir amb quina de les alternatives ens quedem. Són els anomenats *Principis avaluatius*.

2 Cohesió

Definició. Cohesió

Mesura qualitativa sobre el grau de relació i focalització de les tasques o responsabilitats assignades a un mateix component.

Aproximació quantitatives a la cohesió des dels components. La cohesió és un concepte qualitatiu sense un correlat quantitatiu clar. Una mesura quantitativa, totalment arbitrària, que pot donar informació sobre la cohesió, és el nombre de tasques o responsabilitats diferents assignades a un mateix component: com més responsabilitats menys cohesió.

Aproximació quantitatives a la cohesió des dels objectes. Una mesura quantitativa alternativa és el nombre de missatges diferents que reben els objectes realització d'un mateix component: com més missatges menys cohesió. Aquesta mesura quantitativa, en els diagrames de col·laboració (DC), es correspon al nombre de sagetes amb etiqueta diferent que rep un mateix component (cal però comptar *tots* els DC: l'àmbit d'anàlisi és tot el sistema).[→]

2.4.3. Una densa i fràgil teranyina de decisions, pàgina 27

Problemes d'una cohesió baixa. Un component poc cohesionat fa moltes coses. Com a conseqüència el seu disseny és complex, i el manteniment difícil. Una alta cohesió, per contra, significa un disseny més simple, i una millor mantenibilitat i extensibilitat.

Alta cohesió (GRASP). Davant de dues alternatives d'assignació de responsabilitats, ens decidim per aquella que en el disseny resultant els components tenen una cohesió més elevada.

Principi



3 Acoblament

Definició. Acoblament

Mesura qualitativa de la força amb què un component està connectat o té coneixement d'altres components.

L'acoblament és coneixement. L'acoblament mesura, qualitativament, el coneixement que un component necessita del seu entorn.

Aproximació quantitativa a l'acoblament. L'acoblament és un concepte qualitatiu sense un correlat quantitatiu clar. Una mesura quantitativa, totalment arbitrària, que pot donar informació sobre l'acoblament, és el nombre de dependències que té un component. En el MComp es correspon al nombre de sagetes sortints del component (cal comptar les visibilitats d'atribut, però també les locals i de paràmetre).¹

Problemes de l'alt acoblament. L'acoblament és coneixement. La comprensió del comportament o del disseny d'un component altament acoblat es fa gairebé impossible sense analitzar els components als quals s'acobla. Si l'acoblament és baix podem estudiar el component aïlladament; si l'acoblament és alt l'estudi s'ha d'ampliar als components acoblats.

Excursió. (Acoblament i implementació) En la implementació l'acoblament també té conseqüències. Donades dues unitats d'implementació acoblades, la modificació d'una d'elles genèralment implica, com a mínim, la recompilació de l'altra.

Principi



Baix Acoblament (GRASP). Davant de dues alternatives d'assignació de responsabilitats, ens decidim per aquella que en el disseny resultant els components tenen un acoblament més baix.

4 Anàlisi de l'Encarrilament

Encarrilament i Alta Cohesió. El principi d'Encarrilament assegura una *Alta Cohesió* dels controladors. En aplicar l'Encarrilament la responsabilitat dels controladors queda reduïda a la captura de les peticions fetes pels usuaris del sistema (els ES); les responsabilitats pròpies de cada ES el controlador les *delega* al component més adient. En aquest sentit, el principi d'Encarrilament es pot veure com un cas particular del principi de l'*Alta Cohesió*.

¹Un nou argument per defensar la presència de les dependències a MComp, tot i que això signifiqui violar *Espill*, és que d'aquesta manera es pot trobar a partir de MComp una aproximació quantitativa a l'acoblament del model considerat. Aquest argument reforça els arguments presentats a l'apartat 4.2.2. Dependències, i a l'apartat 4.4. Consistència dels models. →, →

Encarilament i Baix Acoblament. És evident que per poder delegar aquestes responsabilitats necessita estar acoblat amb d'altres components del domini. Però, a no ser que el controlador resolgui totes les responsabilitats sense col·laborar amb ningú, es tracta d'acoblements que segurament també s'haurien de mantenir encara que violéssim el principi de l'*Encarrilament*.

Principis i definicions del capítol

Principis

Alta cohesió (*GRASP*), 127

Baix Acoblament (*GRASP*),
128

Definicions

Acoblament, 128

Cohesió, 127

Capítol 9

Una patata que crema

1	<i>GLS</i> Tasca pendent	135
2	El problema de la patata calenta	135
2.1	La patata calenta	135
2.2	Anàlisi comparativa de les diferents actituds	135
2.3	Conclusions sobre la patata calenta	138
3	Ús, emmagatzematge i creació	140
3.1	Què ens en diu MC	140
3.2	Ús	140
3.3	Emmagatzematge	141
4	Repositoris	147
	Definicions	150

Contingut detallat del capítol 9

1	<i>GLS</i> Tasca pendent	135
	On som	135
	Nova perspectiva	135
2	El problema de la patata calenta	135
2.1	La patata calenta	135
	La patata calenta	135
	Possibilitats per considerar	135
2.2	Anàlisi comparativa de les diferents actituds	135
2.2.1	Context de l'anàlisi	135
	Peticionari, creador i objectiu	135
	Petició i visibilitat	135
	Acoblaments en l'emmagatzematge	135
	Acoblaments en l'ús i en la comunicació	136
	Acoblaments en la creació	136
	Patata amb una única finalitat	136
	Anàlisi que queda com a exercici	136
2.2.2	Emmagatzematge	136
	Context	136
	Localitat de la creació	136
	Exigència de la no localitat en l'emmagatzematge	136
	Dependència entre <i>creador</i> i <i>objectiu</i>	136
	Anàlisi de la cohesió	136
	Anàlisi de l'acoblament	136
2.2.3	Ús	137
	Context	137
	Anàlisi de la cohesió	137
	Anàlisi de l'acoblament	137
2.2.4	Comunicació endavant	137
	Context	137
	Anàlisi de la cohesió	137
	Anàlisi de l'acoblament	137
2.2.5	Comunicació enrera	137
	Context	137
	Anàlisi de la cohesió	137
	Anàlisi de l'acoblament	137
2.3	Conclusions sobre la patata calenta	138
	Cohesió	138
	Acoblament	138
	Prohibició de creació només comunicativa	138
2.3.1	La patata calenta i els diferents rols	138
	Rols i responsabilitats	138
	Rols sobre la patata calenta	138
	La patata calenta en termes de rols	138

3 Ús, emmagatzematge i creació	140
3.1 Què ens en diu MC	140
Error freqüent	140
Responsabilitats del disseny	140
3.2 Ús	140
Ús	140
Creació d'ús	140
3.3 Emmagatzematge	141
3.3.1 Creació amb emmagatzematge	141
3.3.2 Col·laboració i emmagatzematge	141
Context de l'anàlisi	141
Col·laboració per a l'emmagatzematge	141
Visibilitats per a l'emmagatzematge	141
Col·laboracions amb l'objecte emmagatzemat	141
3.3.3 Ús de l'objecte emmagatzemat	141
Petició d'ús	141
Emmagatzematge múltiple	141
Problemes de visibilitat	142
Una solució errònia	142
Necessitat de la visibilitat d'atribut	142
Emmagatzematge simple i emmagatzematge múltiple	142
3.3.4 Recuperació de l'objecte emmagatzemat	143
Petició de recuperació en un emmagatzematge monoavaluat	143
Emmagatzematge	143
Visibilitat d'emmagatzematge	143
Petició d'emmagatzematge	143
Multiplicitat de l'emmagatzematge	143
Petició de recuperació	143
Petició d'us	143
Petició de recuperació en un emmagatzematge multiavaluat	143
Informació addicional en els emmagatzematges multiavaluats	144
Emmagatzematge	144
Visibilitat d'emmagatzematge	144
Petició d'emmagatzematge	144
Multiplicitat de l'emmagatzematge	144
Petició de recuperació	144
Petició d'ús	144
Problemes de visibilitat	145
Necessitat de la visibilitat d'atribut	145
3.3.5 El mecanisme de l'emmagatzematge	145
Com s'emmagatzema	145
Emmagatzematge múltiple	146
Visibilitats per a l'emmagatzematge	146
Emmagatzematge i enllaç d'emmagatzematge	146
Enllaç dirigit i visibilitat d'emmagatzematge	146

4 Repositoris	147
El paper dels multiobjectes: recordatori	147
Repositoris	147
Repositori i emmagatzematge	147
Repositoris i multiobjectes	147
Excursió. (Duplicació de realitzacions)	148
Fragments i distribució	149
Definicions	150

1 ^{GLS} Tasca pendent

On som. En el capítol 7.^{GLS}Cas d'ús *ferInscripció*, pàgina 117[→], hem estudiat la creació d'una nova realització en termes de la informació necessària per crear-la (hem usat el principi *Expert*); el resultat han estat dues alternatives possibles. En el capítol 8.*Anàlisi d'alternatives*, pàgina 125[→], s'han introduït els *principis avaluatius* que permeten avaluar la bondat de les diferents alternatives.

7.^{GLS}Cas d'ús *ferInscripció*, pàgina 117

8.*Anàlisi d'alternatives*, pàgina 125

Nova perspectiva. Abans, però, d'aplicar els *principis avaluatius* a les dues alternatives proposades per *Expert*, estudiarem la problemàtica de la creació d'una nova realització des de la perspectiva de l'ús del nou objecte.

2 El problema de la patata calenta

2.1 La patata calenta

La patata calenta. Suposem que acabem de crear una nova realització. Aquesta nova realització és com una patata calenta que ens crema a les mans: què fem amb ella? Aquest és el problema de la *patata calenta*.

Un cop creat un nou objecte, què fem amb ell?

(16) ?

Possibilitats per considerar. Hem creat un nou objecte, la nostra patata calenta. Tenim tres opcions: ens la mengem (*ús*), ens la quedem (*emmagatzematge*) o la passem (*comunicació*).¹ Els principis de *Baix Acoblament* i *Alta Cohesió* permeten avaluar i comparar els diferents models resultants. Tot seguit en fem l'anàlisi.

2.2 Anàlisi comparativa de les diferents actituds

2.2.1 Context de l'anàlisi

Peticionari, creador i objectiu. Per tal de facilitar els raonaments, considerarem el següent escenari. Un objecte p (el *peticionari*) demana a c (el *creador*) que creï un objecte o (l'*objectiu*).

Petició i visibilitat. Per tal de fer la petició cal que el *peticionari* tingui algun tipus de visibilitat sobre el *creador* ($p \rightarrow c$). En principi no podem suposar cap àmbit visibilitat determinat i per això en els MComp l'expressarem en termes de dependència.

¹Podem fer simultàniament més d'una d'aquestes accions.

Acoblaments en l'emmagatzematge. L'emmagatzematge implica visibilitat d'atribut sobre l'*objectiu*. Això és així perquè justament la funció de l'emmagatzemador és mantenir un enllaç dirigit sobre l'*objectiu* que perduri fora del context d'emissió del missatge de creació.

Acoblaments en l'ús i en la comunicació. L'ús i la comunicació impliquen dependència sobre l'*objectiu*: cal conèixer per poder usar o comunicar.

Acoblaments en la creació. El *creador* està acoblat a l'*objectiu*: com que el crea, necessàriament el coneix ($c \rightarrow o$).

Patata amb una única finalitat. En l'anàlisi que segueix analitzem cadascuna de les tres funcionalitats possibles de la patata calenta: ús, emmagatzematge i comunicació. Per cadascuna d'aquestes funcionalitats fem una anàlisi acurada tot considerant que amb la patata calenta només fem el que diu el supòsit de l'anàlisi.

Anàlisi que queda com a exercici. Evidentment amb una patata podem fer més d'una cosa: primer desar-la i després menajar-la; o primer desar-la i després passar-la a algú altre. Deixem com a exercici les anàlisis dels casos en què la patata calenta té més d'una funcionalitat.

2.2.2 Emmagatzematge

Context. El *creador* rep la responsabilitat de crear un objecte, i un cop creat se'l guarda.

Localitat de la creació. La petició s'ha fet amb un missatge sobre l'objecte *creador*; per tant el nou objecte es crea localment (a la petició de creació) dins el *creador*.

Exigència de la no localitat en l'emmagatzematge. L'emmagatzematge no pot ser una propietat local a la petició de creació, altrament mai no podríem recuperar l'objecte emmagatzemat un cop finalitzada la seva creació.^{2,→}

9.3.3.Emmagatzematge, pàgina 141

Dependència entre *creador* i *objectiu*. De tot plegat resulta que el *creador*, si vol emmagatzemar l'*objectiu* acabat de crear, ha de tenir una dependència no local sobre l'*objectiu*.

Anàlisi de la cohesió. El *creador*, pel que sabem, té dues responsabilitats: la de crear, i la d'emmagatzemar el nou objecte.³

²A l'apartat 9.3.3.Emmagatzematge, pàgina 141, s'analitza en profunditat què significa emmagatzemar i quins són els mecanismes per a l'emmagatzematge.

³Volem avaluar la cohesió del *creador*. Dues responsabilitats no són moltes, a priori. Caldria analitzar si les dues responsabilitats estan molt relacionades entre si (la qual cosa ens donaria una cohesió alta) o no. De moment no hi entrem, i ens limitem a comptar les responsabilitats assignades.

Anàlisi de l'acoblament. Pel fet de demanar al *creador* que sigui l'*emmagatzemador* del nou objecte creat, no s'ha introduït cap nou acoblament.

2.2.3 Ús

Context. El *creador*, un cop creat el nou objecte (visibilitat local al mètode de creació) li envia un missatge i el bandeja (altrament estaríem comunicant o emmagatzemant el nou objecte). Per tant el *creador* té visibilitat local sobre l'*objectiu*.

Anàlisi de la cohesió. El *creador*, pel que sabem, té dues responsabilitats: la de crear i la d'usar el nou objecte.

Anàlisi de l'acoblament. En termes d'acoblament la situació actual és molt similar a la de l'emmagatzematge, amb la diferència que allà on hi havia visibilitat no local aquí hi ha una visibilitat local, i usem enlloc d'emmagatzemar.

2.2.4 Comunicació endavant

Context. En la *comunicació endavant* passem el nou objecte com a argument d'un missatge enviat a un objecte x (*receptor*) del qual el mètode *creador* en té visibilitat.⁴ Per tant el *creador* té visibilitat local sobre l'*objectiu*.

Anàlisi de la cohesió. El *creador*, pel que sabem, té dues responsabilitats: la de crear i la de comunicar el nou objecte al receptor.

Anàlisi de l'acoblament. El *receptor* queda acoblat a l'*objectiu* (en té visibilitat de paràmetre); el *creador* ha d'estar acoblat al *receptor* (n'ha de tenir visibilitat per poder-li enviar l'*objectiu*).⁵

2.2.5 Comunicació enrera

Context. En la *comunicació enrera*, el nou objecte creat es retorna a l'objecte *peticionari*. Per tant el *creador* té visibilitat de paràmetre sobre l'*objectiu*; i el *peticionari* en té algun tipus de visibilitat.⁶

Anàlisi de la cohesió. El *creador*, pel que sabem, té dues responsabilitats: la de crear i la de comunicar el nou objecte al *peticionari*.

⁴Aquí no podem fer cap suposició sobre la visibilitat que el *creador* té sobre el receptor: els tres casos de visibilitat són possibles.

⁵En principi res no impedeix que la comunicació sigui cap el mateix objecte *creador*. En aquest cas *receptor* i *creador* coincideixen. Si el missatge de comunicació és públic llavors el *creador* té visibilitat de paràmetre sobre l'*objectiu*. Si el missatge de comunicació és privat aquest el podem considerar com una factorització de la tasca que ha de realitzar el mètode de creació de l'*objectiu*; en aquest cas cal analitzar el mètode en la seva totalitat (cal desfer les factoritzacions) per veure què estem fent exactament: si comunicant, usant o emmagatzemant.

⁶En principi qualsevol de les tres visibilitats és possible.

Anàlisi de l'acoblament. El *peticionari* queda acoblat a l'*objectiu*. De fet el *creador* només ha fet de pont per establir aquest lligam.⁷

2.3 Conclusions sobre la patata calenta

Cohesió. En tots els casos sembla (en una anàlisi superficial) que el *creador* té una cohesió similar (dues responsabilitats).

Acoblament. En la comunicació enrera el *peticionari* està més acoblat que en els altres casos; en la comunicació endavant, per mor de l'existència de l'objecte receptor, el *creador* té un acoblament superior al dels altres casos. →

Prohibició de creació només comunicativa. Per tant:

(18)

- El principi del *Baix acoblament* prohibeix els casos de creació amb l'única finalitat de comunicar l'objecte acabat de crear
- La creació d'una nova realització exigeix usar-la o emmagatzemar-la

2.3.1 La patata calenta i els diferents rols

Rols i responsabilitats. Sovint per expressar que un objecte $a:A$ té una responsabilitat *resp*, direm que $a:A$ té el *rol resp*. Així, tenir un rol és sinònim d'exercir una responsabilitat

Rols sobre la patata calenta. Si A crea B direm que A és el *creador* de B (o bé que en té el rol). Si A usa B direm que A és *usuari* de B (o bé que en té el rol). Si A emmagatzema B direm que A és *emmagatzemador* de B (o bé que en té el rol).

La patata calenta en termes de rols. L'anàlisi de la *patata calenta* diu que el rol de *creador* ha de coincidir en un mateix objecte amb el rol d'*usuari* o amb el rol d'*emmagatzemador*.

⁷En principi res no impedeix que la comunicació sigui cap el mateix objecte *creador*. En aquest cas *peticionari* i *creador* coincideixen. Si el missatge de comunicació és públic (i en el cas de la comunicació enrera aquest missatge només pot ser el mateix missatge de creació) es manté la visibilitat de paràmetre del *peticionari* sobre l'*objectiu*. Si el missatge de comunicació és privat aquest el podem considerar com un factor d'una tasca més general que ha de realitzar algun mètode del *peticionari*; en aquest cas cal analitzar en la seva totalitat (cal desfer les factoritzacions) el mètode públic del *peticionari* que acaba demanant la creació de l'*objectiu* per veure què estem fent exactament: si comunicant, usant o emmagatzemant.

Mostrem els *acoblaments mínims* pel problema de la patata calenta.

Cal observar que es presenten els acoblaments en forma d'enllaç dirigit monoavaluat. De fet però la multiavaluació és possible tant a $p \rightarrow c$ com a $c \rightarrow r$.

Emmagatzematge	$p \rightarrow c$ $c \xrightarrow{atr} o$	Visibilitat d' <i>atribut</i> sobre l'objectiu
Ús	$p \rightarrow c$ $c \xrightarrow{l} o$	Visibilitat <i>local</i> sobre l'objectiu
Comunicació endavant	$p \rightarrow c$ $c \xrightarrow{l} o$ $c \rightarrow r$ $r \xrightarrow{p} o$	<ul style="list-style-type: none"> • Apareix un nou component, el <i>receptor</i> • Calen visibilitats cap i des del receptor
Comunicació enrera	$p \rightarrow c$ $c \xrightarrow{p} o$ $p \rightarrow o$	El <i>peticionari</i> queda acoblat a l' <i>objectiu</i>

Llegenda

p: Peticionari c: Creador
o: Objectiu r: Receptor

Figura 9.1: Acoblaments de la patata calenta

3 Ús, emmagatzematge i creació

3.1 Què ens en diu MC

Error freqüent. Un error freqüent és considerar que les associacions del MC representen emmagatzematge o ús.

Responsabilitats del disseny. En realitat és el disseny qui decideix quines associacions cal modelitzar i com; és el disseny qui decideix qui usa qui, i qui emmagatzema qui.

(19)

Entre les diferents responsabilitats del disseny hi ha les de *decidir*:

- Quines associacions es modelitzen
- Qui usa qui
- Qui emmagatzema qui

3.2 Ús

Ús. En el disseny una realització $a:A$ *usa* una realització $b:B$ quan per resoldre la responsabilitat demanada el dissenyador decideix que $a:A$ envii un missatge a $b:B$.

Creació d'ús. En alguns casos el disseny d'una petició exigeix la creació d'una nova realització, amb l'única finalitat de fer-ne un ús immediat. És a dir, com a resposta a un determinat missatge, algun objecte envia una petició de creació, seguida d'una petició sobre l'objecte acabat de crear; i acabada aquesta segona petició, el nou objecte deixa de tenir sentit. En aquests casos diem que la creació és una *creació d'ús*, sense necessitat d'emmagatzematge.⁸

Definició. Creació d'ús

Creació d'un objecte per al seu ús immediat. No hi ha cap necessitat d'emmagatzematge.

⁸En la realitat el que hi ha és un emmagatzematge local dins el context d'emissió del missatge de creació: dins d'aquest context es fa la creació i l'ús. En el cas d'implementacions OOP, usaríem una variable local al mètode que respon el missatge que exigeix la creació i l'ús.

3.3 Emmagatzematge

3.3.1 Creació amb emmagatzematge

En la majoria dels casos quan en el disseny es decideix crear una nova realització és perquè s'en preveu un ús futur.⁹ Això exigeix emmagatzemar la nova realització fins el moment d'aquest ús.¹⁰ En aquests casos parlem de *Creació amb emmagatzematge*.

3.3.2 Col·laboració i emmagatzematge

Context de l'anàlisi. Sigui un objecte p (el *peticionari*) que demana a un objecte e (l'*emmagatzemador*) l'emmagatzematge d'un objecte o (l'*objectiu*).

Col·laboració per a l'emmagatzematge. Per tal de fer una petició d'emmagatzematge, el *peticionari* enviarà un missatge a l'*emmagatzemador* tot indicant quin és l'objecte que cal emmagatzemar (l'*objectiu*): $e \cdot \text{desa}(o)$.

Visibilitats per a l'emmagatzematge. Per tant, per tal de poder demanar l'emmagatzematge cal que el *peticionari* tingui algun tipus de visibilitat sobre l'*emmagatzemador* (per poder-li enviar la petició), i cal que l'*emmagatzemador* tingui visibilitat de paràmetre sobre l'*objectiu* (l'*objectiu* és un argument de la petició d'emmagatzematge).¹¹

Col·laboracions amb l'objecte emmagatzemat. La semàntica de l'emmagatzematge és la previsió d'un ús futur. Així, en un futur algun objecte x que tingui visibilitat sobre l'*emmagatzemador*¹² li enviarà o bé un missatge d'ús de l'objecte emmagatzemat ($e \cdot \text{usa}()$), o bé un missatge de recuperació ($o := e \cdot \text{recupera}()$) per tal que un cop recuperat l'objecte emmagatzemat aquest pugui ser usat per $x(o \cdot \text{usa}())$.

3.3.3 Ús de l'objecte emmagatzemat

Petició d'ús. En el cas d'enviar a l'*emmagatzemador* una petició d'ús de l'objecte emmagatzemat ($e \cdot \text{usa}()$), el que estem fent és demanar a l'*emmagatzemador* que recuperi *localment* l'*objectiu* emmagatzemat i que sobre ell apliqui la petició d'ús que estem fent. És a dir, estem demanant que usi l'enllaç dirigit que té sobre l'*objectiu* per tal de delegar-li la nostra petició.

⁹Aquesta previsió pot ser conseqüència directa del propi disseny (per exemple, les realitzacions actives com a mecanisme per comunicar diferents ES), o bé pot venir exigida per l'especificació (per exemple, en el disseny d'una base de dades l'especificació exigeix poder recuperar totes les realitzacions introduïdes).

¹⁰En el cas d'un ús únic, el primer ús exigeix bandejar la realització. Generalment, però, els objectes emmagatzemats previsiblement seran usats múltiples cops.

¹¹En el cas que el *peticionari* i l'*emmagatzemador* siguin el mateix objecte la visibilitat de paràmetre sobre l'*objectiu* només és exigible en el cas que la petició d'emmagatzematge sigui una operació visible des de la interfície del *peticionari*.

¹²Res exigeix que aquest objecte sigui el que ha fet la petició d'emmagatzematge.

Emmagatzematge múltiple. En el cas que l'*emmagatzemador* emmagatzemi mes d'un *objectiu* caldrà que la *petició d'ús* incorpori la informació necessària per tal de poder fer la *selecció* de l'*objectiu*, abans defer la *delegació* de la *petició*.

Problemes de visibilitat. En la *petició d'emmagatzematge* ($e.desa(o)$) l'*emmagatzemador* ha creat un enllaç dirigit de paràmetre cap a l'*objectiu*, però un cop acabada la resposta al missatge $desa(o)$ aquest enllaç dirigit deixa de tenir validesa.¹³ Com doncs pot la *petició d'ús* ($e.usa()$) usar aquest enllaç dirigit per delegar la *petició*, si l'enllaç dirigit ja no és vàlid??

Una solució errònia. La solució que es presenta com a òbvia és donar a l'*emmagatzemador* la informació necessària per a tornar a construir l'enllaç dirigit; és a dir, convertir la *petició d'ús* en $e.usa(o)$. D'aquesta manera l'*emmagatzemador* podrà usar el nou enllaç dirigit de paràmetre per fer l'ús demanat. Aquesta solució però és errònia: per tal que tot sigui correcte cal que l'objecte o usat en la *petició d'emmagatzematge* sigui el mateix objecte o que l'usat en la *petició d'ús*. Amb aquesta pretesa solució l'*emmagatzemador* no cal que emmagatzemi res (li oferim tota la informació necessària en cada moment), i en canvi el *peticionari* (o algú altre) s'ha de preocupar de fer l'emmagatzematge de o amb la intenció d'assegurar que en les crides $e.desa(o)$ i $e.usa(o)$ l'objecte o sigui el mateix.

Necessitat de la visibilitat d'atribut. La solució correcta al problema de la visibilitat exposat és adonar-se que la responsabilitat de l'emmagatzematge té un ?mbit superior al local: en un moment determinat es demana l'emmagatzematge; en un moment posterior es demana l'ús. Són dos moments diferents, i per tant dues peticions diferents. En conseqüència l'àmbit ha de ser superior al local; i això significa que cal que l'*emmagatzemador* tingui un enllaç dirigit d'atribut cap a l'*objectiu*; cal una *visibilitat d'atribut* de l'*emmagatzemador* sobre l'*objectiu*.¹⁴ És a dir, demanem que s'usi la dependència de paràmetre $e:E \rightarrow o:o$ per crear un enllaç d'atribut $e:E \xrightarrow{atr} o^*$ tal que $o:o$ formi part de la destinació de l'enllaç dirigit en qüestió.

Emmagatzematge simple i emmagatzematge múltiple. Si l'*emmagatzemador* $e:E$ emmagatzema un únic *objectiu* $o:o$ n'hi ha prou en mantenir l'enllaç dirigit $e:E \xrightarrow{atr} o:o$. Si l'*emmagatzemador* $e:E$ emmagatzema múltiples objectes, per tal d'emmagatzemar un *objectiu* $o:o$ cal mantenir l'enllaç dirigit $e:E \xrightarrow{atr} [\dots, o:o, \dots]$; és a dir, l'enllaç dirigit $e:E \xrightarrow{atr} o^*$ on $o:o$ forma part de la destinació de l'enllaç dirigit.

¹³Cal recordar que un enllaç dirigit de paràmetre té la mateixa validesa que un enllaç dirigit local. El que els diferencia és la comunicació a la interfície de l'objectiu de l'enllaç dirigit.

¹⁴L'argument seguit és: l'àmbit no pot ser local per la semàntica de l'emmagatzematge; l'àmbit de paràmetre comporta problemes; per tant cal un àmbit d'atribut. Alternativament podem veure la visibilitat de paràmetre com un cas particular de la visibilitat local (en la que es publica la destinació de l'enllaç dirigit); i per tant en no admetre la localitat l'única visibilitat possible, si ha de ser d'àmbit superior a la local, és la visibilitat d'atribut.

3.3.4 Recuperació de l'objecte emmagatzemat

Petició de recuperació en un emmagatzematge monoavaluat. Si l'enllaç dirigit que l'emmagatzemador té sobre l'objectiu és un enllaç dirigit monoavaluat, llavors la petició de recuperació d'un objecte emmagatzemat el que fa és demanar a l'emmagatzemador que ens ofereixi, a través de la interfície de l'operació de recuperació, la destinació de l'enllaç dirigit que té sobre l'objectiu.

Exemple 52 (Recuperació en un emmagatzematge monoavaluat)

Emmagatzematge. Supposem una visibilitat $A \xrightarrow{atr} B$. Siguin dos objectes $a : A$ i $b : B$ tals que a és l'emmagatzemador de b .

Visibilitat d'emmagatzematge. L'emmagatzematge és possible perquè hi ha una visibilitat d'atribut que permet mantenir un enllaç dirigit d'atribut $a : A \xrightarrow{atr} b : B$.¹⁵

Petició d'emmagatzematge. La petició d'emmagatzematge $a . desa(b)$ el que fa és demanar a l'objecte $a : A$ que usi la visibilitat $A \xrightarrow{atr} B$ per crear l'enllaç dirigit d'atribut $a : A \xrightarrow{atr} b : B$, on la destinació de l'enllaç dirigit és justament l'objecte donat com a argument. És a dir, demanem que s'usi la dependència de paràmetre $a : A \rightarrow b : B$ per crear l'enllaç dirigit d'atribut $a : A \xrightarrow{atr} b : B$.

Multiplicitat de l'emmagatzematge. La multiplicitat de la visibilitat $A \rightarrow B$ el que ens diu és que cada objecte $a : A$ té permís per emmagatzemar com a molt un objecte $b : B$.

Petició de recuperació. Per recuperar l'objecte emmagatzemat caldrà una petició de recuperació $o := a . recupera()$. Aquesta petició el que fa és demanar que es publiqui la destinació de l'enllaç dirigit d'atribut que és una realització de la visibilitat $A \xrightarrow{atr} B$ i tal que el seu origen és el propi objecte a .

Petició d'ús. La petició d'ús $a . usa()$ el que fa és usar l'enllaç dirigit $a : A \xrightarrow{atr} b : B$ per delegar la petició sobre l'objecte emmagatzemat: $b . usa()$.

Petició de recuperació en un emmagatzematge multiavaluat. Si l'enllaç dirigit que l'emmagatzemador té sobre l'objectiu és un enllaç dirigit multiavaluat, llavors la petició de recuperació d'un objecte emmagatzemat el que fa és demanar a l'emmagatzemador que ens ofereixi, a través de la interfície de l'operació de recuperació, un objecte determinat d'entre tots els que són destinació de l'enllaç dirigit usat per l'emmagatzematge. És a dir, la petició de recuperació

¹⁵Ja hem donat alguns arguments per la necessitat d'una visibilitat d'atribut per a l'emmagatzematge, tot i que encara no hem arribat a la conclusió ferma d'aquesta necessitat (vegeu la conclusió 20., pàgina 146, i la conclusió 21, pàgina 147). \rightarrow, \rightarrow

crea un enllaç dirigit monoavaluat entre el peticionari i l'objecte recuperat de l'emmagatzemador.

Informació addicional en els emmagatzematges multiavaluats. En cas d'un emmagatzematge multiavaluat la petició de recuperació ha d'incloure informació addicional que indiqui a l'emmagatzemador quin dels molts objectes emmagatzemats és el que es vol recuperar; és a dir, ha de d'incloure la informació necessària pel missatge de selecció. Generalment aquesta informació addicional és l'identificador de l'objecte que es vol recuperar: $o := e.recupera(id)$.

Exemple 53 (Recuperació en un emmagatzematge multiavaluat)

Emmagatzematge. Suposem una visibilitat $A \xrightarrow{atr} B^*$. Sigui dos objectes $a : A$ i $b : B$ tals que a és l'emmagatzemador de b .

Visibilitat d'emmagatzematge. L'emmagatzematge és possible perquè hi ha una visibilitat d'atribut que permet mantenir un enllaç dirigit d'atribut $a : A \xrightarrow{atr} B^*$.¹⁶ L'emmagatzematge el que exigeix és que cal assegurar que $b : B$ sigui un dels objectes que formen part de la destinació d'aquest enllaç dirigit.

Petició d'emmagatzematge. La petició d'emmagatzematge $a.desa(b)$ el que fa és demanar a l'objecte $a : A$ que:

1. Creï l'enllaç dirigit d'atribut $a : A \xrightarrow{atr} B^*$ perm?s per la visibilitat $A \xrightarrow{atr} B^*$, en cas que no existeixi prèviament.
2. Afegeixi l'objecte donat com a argument a la destinació de l'enllaç dirigit $a : A \xrightarrow{atr} B^*$. És a dir, demanem que s'usi la dependència de paràmetre $a : A \rightarrow b : B$ per modificar (o crear si és necessari) l'enllaç dirigit d'atribut $a : A \xrightarrow{atr} B^*$.

Multiplicitat de l'emmagatzematge. La multiplicitat de la visibilitat $A \rightarrow B^*$ el que ens diu és que cada objecte $a : A$ té permís per emmagatzemar tants objectes realització de A com vulgui.

Petició de recuperació. Anomenarem enllaç d'emmagatzematge a aquell enllaç dirigit que és una realització de la visibilitat $A \xrightarrow{atr} B^*$ i tal que el seu origen és l'objecte a . Per recuperar un objecte emmagatzemat caldrà una petició de recuperació $o := a.recupera(id)$. Aquesta petició el que fa és demanar que es publiqui quin és l'objecte, d'entre tots els que formen part de la destinació de l'enllaç d'emmagatzematge, que s'identifica amb el valor id .

Petició d'ús. En aquest cas la petició d'ús ha d'incloure la informació necessària per fer la selecció. Per exemple, $a.usa(id)$ usa l'enllaç dirigit $a : A \xrightarrow{atr} B^*$ per seleccionar l'objecte $b : B$ que per identificador té el valor id , i delegar-li la petició.

Problemes de visibilitat.¹⁷ En la petició d'emmagatzematge (`e.desa(o)`) l'*emmagatzemador* ha creat (o modificat) un enllaç dirigit de paràmetre cap a l'*objectiu*, però un cop acabada la resposta al missatge `desa(o)` aquest enllaç dirigit deixa de tenir validesa.¹⁸ Com doncs pot la petició de recuperació (`o:=e.recupera()`) usar aquest enllaç dirigit per comunicar la seva destinació?

Necessitat de la visibilitat d'atribut. La solució correcta al problema de la visibilitat exposat és adonar-se que la responsabilitat de l'emmagatzematge té un àmbit superior al local: en un moment determinat es demana l'emmagatzematge; en un moment posterior es demana l'ús. Són dos moments diferents, i per tant dues peticions diferents. En conseqüència l'àmbit ha de ser superior al local; i això significa que cal que l'*emmagatzemador* tingui un enllaç dirigit d'atribut cap a l'*objectiu*; cal una *visibilitat d'atribut* de l'*emmagatzemador* sobre l'*objectiu*.¹⁹

3.3.5 El mecanisme de l'emmagatzematge

Com s'emmagatzema. La petició d'emmagatzematge (`e.desa(o)`) crea un enllaç dirigit de paràmetre entre l'*emmagatzemador* i l'*objectiu*. La resposta de la petició d'emmagatzematge significa assegurar que la destinació d'aquest enllaç dirigit de paràmetre forma part de la destinació de l'enllaç dirigit d'atribut que té l'*emmagatzemador* com a origen, i com a destinació un conjunt d'objectes que són del mateix component que l'*objectiu* en qüestió.

Exemple 54 (Emmagatzematge monoavaluat) *Un codi típic en Java és el següent:*

```
class Emmagatzemador
{
    Objectiu desat; // Enllaç dirigit que manté
                  // l'emmagatzematge

    void desa(Objectiu o)
    {
        desat = o;
    }
}
```

¹⁶Ja hem donat alguns arguments per la necessitat d'una visibilitat d'atribut per a l'emmagatzematge, tot i que encara no hem arribat a la conclusió ferma \rightarrow, \rightarrow d'aquesta necessitat.

20, pàgina 146

¹⁷Aquest paràgraf i el següent són idèntics als equivalents de l'apartat sobre la petició d'ús, adaptats ara a la petició de recuperació. No aporten res de nou, però s'inclouen per completesa expositiva.

21, pàgina 147

¹⁸Cal recordar que un enllaç dirigit de paràmetre té la mateixa validesa que un enllaç dirigit local. El que els diferencia és la comunicació a la interfície de l'objectiu de l'enllaç dirigit.

¹⁹L'argument seguit és: l'àmbit no pot ser local per la semàntica de l'emmagatzematge; l'àmbit de paràmetre comporta problemes; per tant cal un àmbit d'atribut. Alternativament podem veure la visibilitat de paràmetre com un cas particular de la visibilitat local (en la que es publica la destinació de l'enllaç dirigit); i per tant en no admetre la localitat l'única visibilitat possible, si ha de ser d'àmbit superior a la local, és la visibilitat d'atribut.

```
Objectiu recupera()  
{  
    return desat;  
}  
  
void usa()  
{  
    desat.usa();  
}  
}
```

Emmagatzematge múltiple. Com hem vist, en el cas que un mateix emmagatzemador hagi d'emmagatzemar més d'un objecte simultàniament les coses no varien. L'única diferència és que ara tant l'ús com la recuperació s'hauran de combinar amb un missatge de selecció.

Visibilitats per a l'emmagatzematge. L'emmagatzemador ha de tenir visibilitat de paràmetre sobre l'objectiu en les operacions d'emmagatzematge i de recuperació. Per tal que la recuperació o les peticions d'ús siguin possibles cal que l'emmagatzemador tingui visibilitat d'atribut sobre l'objectiu. Així:

(20)

L'emmagatzematge exigeix que l'emmagatzemador tingui visibilitat d'atribut sobre l'objectiu:

$$\text{Emmagatzemador} \xrightarrow{\text{atr}} \text{Objectiu}^*$$

Definició. Enllaç (dirigit) d'emmagatzematge

Donat un objecte e emmagatzemador d'un objecte o , anomenarem enllaç (dirigit) d'emmagatzematge de o , a l'enllaç dirigit d'atribut $e : \text{Emmagatzemador} \xrightarrow{\text{atr}} \text{Objectiu}^*$ tal que la destinació de l'enllaç dirigit conté l'objecte o considerat.

Emmagatzematge i enllaç d'emmagatzematge. L'emmagatzematge exigeix un enllaç dirigit d'atribut des de l'emmagatzemador a l'objectiu de l'emmagatzematge; l'enllaç (dirigit) d'emmagatzematge és justament aquest enllaç dirigit.

Definició. Visibilitat d'emmagatzematge

Anomenarem visibilitat d'emmagatzematge la visibilitat d'atribut $\text{Emmagatzemador} \xrightarrow{\text{atr}} \text{Objectiu}^*$ exigida per l'emmagatzematge.

Enllaç dirigit i visibilitat d'emmagatzematge. L'enllaç d'emmagatzematge és la realització de la visibilitat d'emmagatzematge; la visibilitat d'emmagatzematge

matge és la potencialitat que permet l'existència de l'enllaç dirigit d'emmagatzematge.

Tot objecte o *objectiu* d'un *emmagatzematge* ha de tenir un *emmagatzemador* e tal que:

- *L'emmagatzemador és origen de l'enllaç dirigit d'emmagatzematge*

Existeix un enllaç dirigit d'emmagatzematge que té com a origen l'emmagatzemador:
 $e : \text{Emmagatzemador} \xrightarrow{\text{atr}} \text{Objectiu}^*$

- *L'objectiu és emmagatzemat*

L'objectiu ha de formar part de la destinació d'aquest enllaç dirigit:

$e : \text{Emmagatzemador} \rightarrow [x_1, \dots, o, \dots, x_k]$

(21)

4 Repositoris

El paper dels multiobjectes: recordatori. Els multiobjectes són una ficció que permeten treballar còmodament amb les visibilitats multiavaluades; en concret simplifiquen el mecanisme de col·laboració a través dels enllaços dirigits multiavaluats: els missatges de grup no traspassen la frontera del multiobjecte, els missatges d'enllaç arriben a tots i cadascun dels objectes que formen part del multiobjecte.

Repositoris. Quan es parla d'emmagatzematge interessa raonar sobre quins objectes són responsabilitat d'un mateix emmagatzemador; és a dir, interessa raonar sobre quin és el conjunt d'objectes que són l'objectiu dels *enllaços dirigits d'emmagatzematge*. D'aquí el concepte de *repositori*:

Definició. Repositori

Un repositori^{20, →} de realitzacions del component C (o, abreviadament, repositori de C) és un multiobjecte que emmagatzema realitzacions del component C .

??, pàgina ??

Repositori i emmagatzematge. La definició de *repositori* exigeix emmagatzematge; és a dir, l'existència d'una visibilitat d'atribut $x \xrightarrow{\text{atr}} C^*$ per algun component x . En la definició de *repositori*, però, no es té en compte qui és l'emmagatzemador (x no apareix a la definició).

²⁰A ??, pàgina ??, reformulem aquesta definició.

Repositoris i multiobjectes. Els multiobjectes permeten estudiar la col·laboració; els repositoris, l'emmagatzematge. Ara bé, com que el mecanisme de l'emmagatzematge és una *visibilitat d'atribut*, el multiobjecte emprat per estudiar la visibilitat d'emmagatzematge és justament el repositori. El resultat són dues visions d'una mateixa idea: el conjunt d'objectes visibles des de l'emmagatzemador.

(22)

- Els *multiobjectes* són una eina per a l'anàlisi i tractament de la *col·laboració*
- Els *repositoris* són una eina per a l'anàlisi i tractament de l'emmagatzematge

Definició. Realització compartida

Donada una realització d'un component direm que és una realització compartida si pertany a més d'un repositori.

Excursió. (Duplicació de realitzacions) La realització compartida significa una *compartició lògica*: una mateixa realització és accessible des de dos repositoris diferents. Un altre concepte és el de la *duplicació de realitzacions* en el que diferents còpies d'una mateixa realització s'emmagatzemen en repositoris diferents. La *duplicació de realitzacions* s'usa, per exemple, quan els diferents repositoris on volem que aparegui una mateixa realització lògica estan en màquines diferents.

En el que segueix no contemplem mai la possibilitat de duplicar una realització.

Definició. Repositori centralitzat

Quan totes les realitzacions del component C existents en el sistema s'emmagatzemen en un mateix repositori, direm que tenim un repositori centralitzat.

Definició. Repositoris disjunts

Donats dos repositoris direm que són disjunts si i només si no hi ha cap realització que pertanyi simultàniament als dos repositoris.

Definició. Repositori distribuït

Quan totes les realitzacions del component C existents en el sistema s'emmagatzemen en repositoris diferents disjunts, direm que la unió de tots aquests

*repositoris és un repositori distribuït de C.*²¹

Definició. Fragment d'un repositori

Cadascun dels repositoris disjunts que emmagatzemen les diferents realitzacions d'un mateix component C és un fragment de repositori de C.

Fragments i distribució. Els repositoris centralitzats tenen un sol fragment: el mateix repositori. Els repositoris distribuïts consten de diferents fragments, i aquests són repositoris disjunts entre ells (és a dir, no hi ha cap realització compartida en dos fragments diferents).²²

²¹Si mantenim la definició actual de repositori el terme *repositori distribuït* és erroni: fa referència a un conjunt de multiobjectes, i no pas a un multiobjecte. Més endavant → refarem la definició de *repositori* i aquesta incongruència desapareixerà.

²²Amb la definició actual de *repositori*, els fragments són repositoris. Quan parlem de fragments és perquè volem fer èmfasi en el fet que no es tracta d'un repositori centralitzat, i que per tant hi ha d'altres repositoris que emmagatzemen realitzacions del del mateix component C que no estan emmagatzemades en el repositori (fragment) que estem considerant.

Principis i definicions del capítol

Definicions

Creació d'ús, 140

Enllaç (dirigit) d'emmagatzematge, 146

Fragment d'un repositori, 149

Realització compartida, 148

Repositori, 147

Repositori centralitzat, 148

Repositori distribuït, 148

Repositoris disjunts, 148

Visibilitat d'emmagatzematge, 146

Capítol 10

Principis de creació

1	Principi Creador (GRASP)	158
2	Principi Expert delegat	159
2.1	Delegació de l'expertesa	159
2.2	Delegació de la creació	160
3	Principi Creador orb	161
3.1	Anàlisi dels usuaris	161
3.2	Oblidem els usuaris	163
4	Emmagatzematge	164
4.1	Estudi del MC	164
4.2	Visibilitat d'emmagatzematge	165
4.3	Principi de l'Emmagatzemador selectiu	165
4.4	Principi de l'Emmagatzemador general	169
4.5	Principi de l'Emmagatzemador canònic	172
5	Principi del Creador canònic	173
5.1	Creació i emmagatzematge	173
5.2	Creador canònic	175
6	Emmagatzematge primari	176
6.1	Creador canònic i emmagatzematge primari	176
6.2	Anàlisi del Creador orb	178
7	Principi del Creador generalitzat	182
8	Migració i compartició de realitzacions	183
8.1	Presentació de les migracions	183
8.2	Presentació de les comparticions	183
8.3	Creador canònic, migracions i comparticions	184
8.4	Creador orb pur, migracions i comparticions	185
8.5	Creació i emmagatzematge primari	187
9	Ús dels diferents principis de creació	187
9.1	Jerarquització dels principis de creació	187
9.2	Candidats no considerats amb l'ús del Creador canònic	188
10	La patata calenta i els principis de creació	189
11	Creació i recuperació de la informació	191

11.1 Anàlisi de la recuperació en una creació	191
11.2 Excursió: Recuperació d'informació	192
Principis	197
Definicions	197

Contingut detallat del capítol 10

1 Principi Creador (GRASP)	158
Coincidència de rols	158
La coincidència de rols com a òptim local	158
Ens la juguem a l'òptim local	158
Creador (GRASP)	158
Creador en termes d'objectes	158
Creador en termes de components	158
Cas òptim de Creador	158
Creador i la patata calenta	158
2 Principi Expert delegat	159
Punt de control: on som	159
2.1 Delegació de l'expertesa	159
Cadena de delegacions	159
Delegació de l'expertesa	159
Delegació i informació d'expertesa	159
Múltiples experts	160
2.2 Delegació de la creació	160
Context de delegació	160
Anàlisi del context	160
Acoblaments en la creació directa	160
Acoblaments en la delegació de la creació	160
Delegació i Creador	160
Creador delegat	160
Delegació de l'expertesa	160
Delegació i Creador delegat	161
3 Principi Creador orb	161
3.1 Anàlisi dels usuaris	161
Creació d'ús	161
Usuari present	161
Usuari futur	161
Usuaris i emmagatzemadors	162
Problemàtica detecció dels <i>usuaris futurs</i>	162
Inestabilitat dels <i>usuaris futurs</i>	162
Inestabilitat dels <i>usuaris presents</i>	162
Ineficàcia de l'estudi dels <i>usuaris</i>	162
3.2 Oblidem els usuaris	163
Creador orb	163
Subsumpció d'altres principis	163
Estabilitat de la proposta	163
Cas particular	163
Independència dels usuaris	163
Creador orb en termes de <i>Cohesió</i>	163
4 Emmagatzematge	164

4.1	Estudi del MC	164
	Visibilitat d'atribut: un mecanisme amb dues funcions	164
	Associacions i emmagatzematge	164
	Estabilitat de l'especificació	164
	Inestabilitat del disseny	164
	Eficàcia de l'estudi de MC	164
4.2	Visibilitat d'emmagatzematge	165
	Context de l'anàlisi	165
	Visibilitat $O \xrightarrow{atr} C^*$	165
	Visibilitat $C \xrightarrow{atr} O^*$	165
4.3	Principi de l'Emmagatzemador selectiu	165
	Emmagatzemador selectiu i Espill	165
	Context de l'anàlisi	165
	Primeres observacions	166
	No violació d'Espill a nivell de les visibilitats	166
	No violació d'Espill a nivell dels enllaços dirigits	166
	Una condició repetida	166
	Una condició que cal assegurar	167
	El camí a la inversa	167
	Anàlisi que farem	167
	Anàlisi pas a pas	167
	Conclusions de l'anàlisi	168
	Emmagatzemador selectiu	168
	Emmagatzemador selectiu a nivell de les abstraccions	168
	Emmagatzemador selectiu a nivell de les realitzacions	168
	Emmagatzemador selectiu a nivell de les realitzacions i el MC	168
	Què aconseguim amb aquestes dues condicions	168
	El perquè de l'adjectiu selectiu	169
4.4	Principi de l'Emmagatzemador general	169
	Emmagatzemador selectiu en termes planers	169
	Els objectes seleccionats	169
	Què ens preguntem exactament	170
	Universalitat de la pregunta	170
	Aplicació de l'Emmagatzemador selectiu	170
	Pregunta reformulada	170
	Resposta simple	170
	Simplificació de les condicions	171
	Emmagatzemador general	171
	Principi en termes de components	171
	Una condició innecessària	171
	Una condició implícita	171
4.5	Principi de l'Emmagatzemador canònic	172
	L'Emmagatzemador general no és massa precís	172
	Suprimim la indefinició	172
	Emmagatzemador (canònic)	172
	Emmagatzemador general o canònic	172

5	Principi del <i>Creador canònic</i>	173
5.1	Creació i emmagatzematge	173
	Exigències per a la creació	173
	Exigències per a l'emmagatzematge	173
	Context de l'anàlisi	173
	Emmagatzemador selectiu	173
	Desig d'universalitat	173
	Emmagatzemador general	173
	Problemes de decidir	173
	Emmagatzemador canònic	175
	A punt de solfa	175
5.2	Creador canònic	175
	Creador canònic	175
	Subsumpció d'altres principis	176
	Emmagatzematge consistent amb MC	176
	Estabilitat de la proposta	176
6	Emmagatzematge primari	176
6.1	<i>Creador canònic</i> i emmagatzematge primari	176
	Inversa hipotètica	176
	Excursió. (Visibilitat inversa)	176
	Emmagatzematge primari i emmagatzemador canònic	176
	Presentació alternativa del <i>Creador canònic</i>	177
	Creador canònic	178
	Emmagatzematge i creació	178
	<i>Creador canònic</i> i <i>Espill</i>	178
6.2	Anàlisi del <i>Creador orb</i>	178
6.2.1	Tipificació dels creadors orbs	178
	Camí recorregut	178
	Tipologia de <i>creadors orbs</i>	178
6.2.2	Creador canònic	180
6.2.3	Creador canònic fals	180
	Creador amb <i>emmagatzematge selectiu</i>	180
	Excursió. (Reconsideració de l'emmagatzemador selectiu)	180
	Creador amb emmagatzemador general	180
	Res d'emmagatzemadors canònics falsos	180
	Camins de fugida	180
6.2.4	Creador orb pur	181
	Una proposta vàlida	181
	Obtenció de l'emmagatzematge primari	181
6.2.5	Creador orb enganyós	181
	Cohesió de l'emmagatzemador canònic	181
	Un problema de baixa cohesió	181
6.2.6	Un parell de definicions	181
	Unes definicions que seran útils	181
7	Principi del <i>Creador generalitzat</i>	182
	Creador generalitzat	182

Objectiu del <i>creador generalitzat</i>	182
El guanyador	182
Un perdedor clar	182
L'altre perdedor	182
Un recurs necessari	182
8 Migració i compartició de realitzacions	183
8.1 Presentació de les migracions	183
Concepte de migració	183
Tipus de migracions	183
8.2 Presentació de les comparticions	183
Concepte de compartició	183
Tipus de comparticions	183
8.3 <i>Creador canònic</i> , migracions i comparticions	184
El context de la migració	184
Migració externa	184
Migració interna	184
Comparticions	185
8.4 <i>Creador orb pur</i> , migracions i comparticions	185
L'emmagatzematge primari del <i>Creador canònic</i>	185
Emmagatzemador orb pur	185
Migració externa	186
Migració interna	186
Compartició	186
Comparticions	186
8.5 Creació i emmagatzematge primari	187
Com assegurem l'emmagatzematge primari	187
9 Ús dels diferents principis de creació	187
9.1 Jerarquització dels principis de creació	187
Resistència a trencar la <i>Modelització continua</i>	187
Delegació com a segona opció	187
Els <i>usuaris</i> com a recurs d'emergència	188
Preponderància de <i>Creador canònic</i>	188
Violació del principi d' <i>Igualtat d'oportunitats</i>	188
9.2 Candidats no considerats amb l'ús del <i>Creador canònic</i>	188
Anàlisi de pèrdues	188
Candidats no contemplats pel principi <i>Creador canònic</i>	188
Què perdem amb els <i>usuaris</i>	188
Què perdem amb els <i>experts</i> que no són ni <i>usuaris</i> ni tam- poc emmagatzemadors	188
Què perdem amb els <i>emmagatzemadors no canònics</i>	188
Quan cal recórrer als <i>emmagatzemadors no canònics</i>	189
10 La patata calenta i els principis de creació	189
Expertesa en la creació	189
Comunicar no pot ser un rol bàsic	189
Coincidència de rols	189
Delegació de l'expertesa	189

Independència dels <i>usuaris</i>	190
La independència és <i>Cohesió</i>	190
Emmagatzematge i MC	190
Orientació de les visibilitats	190
Creació i emmagatzematge canònic	190
Principi preponderant	190
Emmagatzemadors no admesos	190
Conseqüències de la violació de la <i>Igualtat d'oportunitats</i>	190
Anàlisi a dos nivells	190
Simplificació de nomenclatura	190
11 Creació i recuperació de la informació	191
11.1 Anàlisi de la recuperació en una creació	191
Informació d'expertesa en la creació	191
Propietats de l'objecte creat	191
Recuperació de l'expertesa de la creació	191
Maneres de recuperar la informació d'expertesa	191
Responsabilitats del disseny	191
11.2 Excursió: Recuperació d'informació	192
11.2.1 La informació en el MC	192
Atributs dels conceptes	192
Atributs i associacions	193
Excursió. (El paper dels atributs)	193
11.2.2 La informació a MComp	194
Atributs que modelitzen atributs	194
Atributs del disseny	194
Visibilitat d'atribut	194
Excursió. (Atributs i visibilitat d'atribut)	194
Propietats atributives	194
11.2.3 Utilitat de la informació en el disseny	195
Utilitat de la informació	195
Utilitat en el disseny	195
11.2.4 Anàlisi de la recuperació	195
Procés d'anàlisi de la recuperació	195
Recuperació dels atributs	195
Recuperació de les destinacions de les visibilitats d'atribut	195
Paràmetres i visibilitat de paràmetre	195
Excursió. (Paràmetres i visibilitat de paràmetre)	196
Principis	197
Definicions	197

1 Principi Creador (GRASP)

9. Una patata que crema, pàgina 131

Conclusió 18, pàgina 138

Coincidència de rols. En l'anàlisi del problema de la patata calenta,[→] hem vist com en termes d'*acoblament* és bo que el rol de *creador* coincideixi amb el rol d'*usuari* o amb el rol d'*emmagatzemador*.^{1, →} De fet l'òptim està en assignar els diferents rols al mínim d'objectes possibles. La coincidència o convergència d'aquests tres rols en dos objectes diferents redueix les necessitats de comunicació i per tant les visibilitats de paràmetre; la coincidència dels tres rols en un sol objecte elimina les necessitats comunicatives i les visibilitats de paràmetre.²

La coincidència de rols com a òptim local. La coincidència de *creador*, *usuari* i *emmagatzemador* redueix l'*acoblament* de la petició que estem dissenyant; però res diu de l'*acoblament* del disseny complet; ni tampoc de si es violen o no d'altres principis de disseny.^{3, →}

2.4.1. Com assignar una responsabilitat, pàgina 25

Ens la juguem a l'òptim local. La recomanació de fer coincidir els rols és una aposta: ens la juguem a què l'òptim local és la millor opció de cara al disseny complet; ens la juguem a què l'òptim local ens portarà a l'òptim global. Aquesta aposta és la que expressa el següent principi:

Principi



Creador (GRASP). La responsabilitat de crear una realització de B recau en aquella realització d'un component A que en sigui *expert*, i que a més, per altres motius, sigui *usuari* o *emmagatzemador* d'aquesta realització de B.

Creador en termes d'objectes. Recordem que tot i que les responsabilitats les exerceixen els objectes,⁴ l'assignació pròpiament dita es fa sobre els components. En aquest cas però hem presentat el principi *Creador* en termes d'objectes per facilitar-ne la comprensió.

Creador en termes de components. El principi *Creador*, en termes de components, diu que un component A rep la responsabilitat de crear B quan A és expert en B i a més, per altres motius, n'és *usuari* o *emmagatzemador*.

Cas òptim de Creador. En termes d'*acoblament* l'òptim es produeix quan els tres rols, *creador*, *usuari* i *emmagatzemador* coincideixen en un sol component. És a dir, quan la "o" del principi *Creador* es realitza de manera inclusiva.

¹Aquesta afirmació és conseqüència immediata de la prohibició de la creació només comunicativa expressada al requadre 18, pàgina 138.

²L'anàlisi detallada de l'*acoblament* és similar al realitzat en l'anàlisi del problema de la patata calenta, i en concret en l'anàlisi de la comunicació endavant i endarrera. Deixem però aquesta anàlisi com a exercici.

³La no coincidència entre l'àmbit de la decisió i l'àmbit de l'anàlisi són els que ens portaren a plantejar els principis de la *Teranyina* i de la *Reconsideració* contínua; apartat 2.4.1. Com assignar una responsabilitat, pàgina 25.

⁴Recordem que un objecte és la realització d'un component.

Creador i la patata calenta. El principi *Creador* ens diu que el responsable de crear un nou objecte és aquell qui l’ha d’usar o guardar. El que d’alguna manera “prohibeix” el principi és el fet de crear un objecte amb l’única finalitat de comunicar-lo a algú altre.⁵

2 Principi *Expert delegat*

Punt de control: on som. De les tres possibles accions sobre la patata calenta, *Baix acoblament* ha eliminat la comunicació; el resultat queda expressat en el principi *Creador* (GRASP). Queden encara per analitzar dues qüestions:

- Sota quines condicions és possible fer coincidir el rol de *creador* amb els rols de *usuari* i/o *emmagatzemador*
- Com decidim quins són els *usuaris* i els *emmagatzemador* d’una nova realització

2.1 Delegació de l’expertesa

Cadena de delegacions. L’existència d’una *visibilitat indirecta*[→] significa l’existència d’una cadena de visibilitats; i aquesta cadena de visibilitats ofereix un mecanisme per permetre una *cadena de delegacions*, des de qui rep la responsabilitat fins al qui realment la realitza: cada element de la cadena té visibilitat directa del següent element de la cadena, i per tant pot delegar-li la responsabilitat que se li ha encomanat.

Delegació de l’expertesa. Si el primer element d’una cadena de visibilitats és *expert* en alguna tasca, llavors també ho poden ser, per delegació, qualsevol dels elements que formen part de la cadena: n’hi ha prou en passar per paràmetre del missatge de delegació la informació de la qual s’és expert.

Definició. Expert per delegació

Un objecte que rep com a arguments d’un missatge tota la informació necessària per ser expert en una tasca direm que és un expert per delegació sobre aquesta tasca.

Definició. Informació d’expertesa

La informació necessària per a exercir una responsabilitat l’anomenem informació d’expertesa d’aquesta responsabilitat.

⁵*Excursió.* Els patrons de creació (GoF) diuen que sota determinades circumstàncies calen components amb l’única responsabilitat de crear nous objectes que són lliurats a qui ha fet la petició de creació. Els patrons de creació i el principi *Creador* no són antagònics, sinó que es complementen: *Creador* diu qui ha de rebre la responsabilitat; els patrons de creació ens diuen que per exercir aquesta responsabilitat és bo delegar en un component creador, introduït en virtut del principi de la *Fabricació pura*.

Delegació i informació d'expertesa. L'*expert per delegació* és aquell que obté la *informació d'expertesa* d'una responsabilitat del seu propi coneixement i de la informació passada en el moment que algú li *delega* l'exercici de la responsabilitat.

Múltiples experts. La possibilitat de delegar l'expertesa fa que en tot moment poguem tenir múltiples experts: l'*expert directe*, i els *experts per delegació*.

2.2 Delegació de la creació

Context de delegació. Sigui un component A *expert* en la creació de realitzacions de B . Suposem l'existència d'una visibilitat indirecta de $A \rightarrow^* C$, on C és el responsable d'emmagatzemar o usar el nou $b : B$ creat; i siguin C_i els diferents components que componen la cadena de visibilitats (els *intermediaris*). Llavors A pot delegar la creació de B a través de la cadena que passa per tots els C_i i acaba en C : $A \rightarrow C_1 \rightarrow \dots \rightarrow C_k \rightarrow C$.

Anàlisi del context. Donat el context de delegació acabat de presentar, tot seguit analitzarem els acoblaments que es produeixen en els següents supòsits:

- *Creació directa.* L'objectiu $b : B$ el crea el component A ⁶
- *Delegació de la creació.* El component A delega la seva expertesa per tal que finalment l'objectiu el creï C ⁷

Acoblaments en la creació directa. Si l'*expert* (A en el context de creació exposat) *crea* l'objectiu, llavors un cop creat el nou objecte $b : B$ caldrà fer-lo arribar a C (el seu *emmagatzemador* o *usuari*) a través de la cadena de visibilitats directes. El resultat és que tots els *intermediaris* queden acoblats a B . Pel que fa a la *informació de l'expertesa*, aquesta només és coneguda pel *creador* A .

Acoblaments en la delegació de la creació. El resultat de la delegació de la creació és que només C (*emmagatzemador* o *usuari*) queda acoblat a l'*objectiu*, mentre que tots els *intermediaris* queden acoblats a la *informació de l'expertesa*.

Delegació i Creador. La delegació de l'expertesa permet fer coincidir la responsabilitat de creació amb la responsabilitat d'emmagatzemar i/o usar: n'hi ha prou en convertir a l'emmagatzemador i/o a l'usuari en expert per delegació de la creació de l'objectiu; el preu és l'acoblament dels intermediaris amb la informació de l'expertesa.

Principi



Creador delegat. Els candidats a *creador* són els *experts*, si cal *per delegació*, en la creació que a més en són *emmagatzemadors* o *usuaris*

⁶Pròpiament, qui crea l'objectiu és un objecte realització del component A .

⁷Pròpiament, un objecte $: A$ delega la seva expertesa un objecte $: C$, que és qui crea l'objectiu.

Delegació de l'expertesa. El principi del *Creador delegat* fixa les condicions en les que el rol *creador* pot coincidir amb el d'*emmagatzemador* i/o amb el d'*usuari*: en cas que l'*emmagatzemador* i/o a l'*usuari* no siguin *experts* en la creació, cal que es donin les condicions per poder delegar-los-li l'expertesa en la creació.

Delegació i *Creador delegat*. És important fer notar que *Creador delegat* introdueix la delegació de l'expertesa per poder fer convergir els rols de *creador* amb els d'*emmagatzemador* i/o d'*usuari*. La delegació de l'expertesa, però, introdueix l'*acoblament* de tots els *intermediaris* amb la *informació de l'expertesa*, i per això el principi demana la delegació només "si cal". En aquest sentit:⁸

La *delegació* que permet el principi *Creador delegat* cal considerar-la només quan la coincidència de rols no es pugui fer sense la delegació

(23)

3 Principi *Creador orb*

3.1 Anàlisi dels usuaris

Quins són els usuaris de la realització acabada de crear?

(17) ?

Creació d'ús. En aquest cas no hi ha *emmagatzemador*: creem per usar de manera immediata. Per tant, pel principi *Creador delegat* cal que *creador* i *usuari* coincideixin.

Usuari present. Quan en el disseny de la petició que estem considerant algú fa un ús explícit d'una realització direm que aquest algú n'és un *usuari present*.

Usuari futur. Quan en el disseny de la petició que estem considerant ningú no usa la nova realització acabada de crear és perquè estem pensant en un ús futur, la qual cosa ens obliga a l'*emmagatzematge*.⁹ En aquest cas podríem intentar esbrinar quins poden ser els usos futurs, per tal d'esbrinar els possibles *usuaris futurs*. Per fer-ho caldrà *dissenyar en paral·lel* els diferents ES d'un mateix CU (o fins i tot, més d'un CU simultàniament) amb l'objectiu de detectar les

⁸La delegació de l'expertesa acobla els intermediaris amb la informació d'expertesa per a la creació; la creació directa acobla els intermediris amb l'objectiu. El que diu el principi *Creador delegat* és que és preferible l'acoblament amb l'objectiu que amb la informació d'expertesa. Aquí es fa difícil justificar aquesta preferència, que té relació directa amb el principi de la *Cistella*.[→] La idea és que en la comunicació és preferible un objecte complex que molts objectes simples.

⁹Cal observar que l'existència d'usuaris presents no impedeix l'existència d'usuaris futurs. Aquí simplement presentem el cas on clarament hi ha d'haver usuaris futurs, però són molts més els casos possibles.

necessitats comunicatives entre els diferents ES al més aviat possible. Del que es tracta és de conèixer, davant una creació d'una realització, qui, quan i com l'usarà.

Usuaris i emmagatzemadors. El principi *Creador delegat* exigeix que el *creador* coincideixi amb l'*emmagatzemador* i/o l'*usuari*. En la *creació d'ús* no hi ha *emmagatzemador*; en els altres casos sempre hi ha *usuaris*, siguin aquests *presentes* o *futurs*. Per cada possible *usuari* o *emmagatzemador* caldrà analitzar les possibilitats de *delegar-li* l'expertesa en la creació.

Problemàtica detecció dels *usuaris futurs*. L'anàlisi dels possibles *usuaris futurs* és problemàtica. Per una banda complica el disseny en exigir el disseny simultani de diferents ES. Per altra banda aquesta anàlisi introdueix un *acoblament* excessiu entre el disseny de la petició actual, i el disseny de la resta del sistema; àdhuc de millores o ampliacions futures.

Inestabilitat dels *usuaris futurs*. L'anàlisi dels possibles *usuaris futurs* no només és complexa i requereix certes dotes adivinatòries, sinó que el seu resultat és molt inestable: a mesura que avancem en el disseny (o que en fem ampliacions) pot canviar la nostra perspectiva de què ha d'usar què, i en quin moment.

Inestabilitat dels *usuaris presents*. La diferència entre *usuari present* i *usuari futur* la dona el disseny. Però és molt normal que durant el desenvolupament un ús present es converteixi en un ús futur, o a la inversa.

Exemple 55 (Inestabilitat dels *usuaris presents*) *Suposem el context d'un terminal punt de venda (TPV). Inicialment podem haver decidit que en crear la línia de venda (que conté l'article venut i la quantitat d'ítems venuts) no hem de fer res amb ella; només emmagatzemar-la. Més endavant ens n'adonem que aquest és un bon moment per capturar estadístiques de consum del client. I per fer-ho el que fem és enviar la línia de venda acabada de crear al gestor estadístiques per tal que la usi per treure'n la informació rellevant. Inicialment la línia de venda no tenia usuaris presents; en la solució final gestor estadístiques n'és un usuari present.*

Més endavant ens adonem que per resoldre la responsabilitat d'un determinat ES ens cal recórrer al final del dia totes les línies de venda del dia. Llavors podem decidir aprofitar aquest recorregut per generar les estadístiques de consum. Així, com a conseqüència d'una decisió posterior, reconsiderem la nostra decisió i fem que línia de venda torni a no tenir usuaris presents.

Ineficàcia de l'estudi dels *usuaris*. Per tant, si volem construir un principi general per assignar la responsabilitat de creació d'una nova realització, el recurs als possibles usuaris (presentes o futurs) d'aquesta realització no és una bona idea, per mor de la inestabilitat del resultat (i de la difícil detecció en el cas dels usuaris futurs)

3.2 Oblidem els usuaris

Creador orb. Els candidats a *creador* són els *experts*, si cal per per delegació, de la creació que a més són *emmagatzemadors*.

En el cas de *creació d'ús* (creació sense emmagatzematge) els candidats a *creador* són els *experts*, si cal per delegació, de la creació que a més en són *usuaris*.

Principi



Definició. Creador orb

Els candidats a creador proposats pel principi Creador orb reben el nom de creadors orbs.

Subsumció d'altres principis. El principi del *Creador orb* captura les restriccions del principi *Creador delegat*, així com la observació que tota assignació de responsabilitat basada en els *usuaris* és força inestable.

Estabilitat de la proposta. L'obligació de fer coincidir el *creador* amb algun *emmagatzemador* evita la problemàtica de detectar un conjunt estable d'*usuaris*.

Cas particular. El cas particular de la *creació d'ús* permet tractar el cas en què no hi hagi *emmagatzemadors*.

Independència dels usuaris. El principi *Creador orb* ofereix els possibles *creadors* sense fer esment als usuaris, presents o futurs (amb l'excepció del cas particular de la *creació d'ús*). En aquest sentit és *orb*, cec: el *creador* es despreocupa del context i només té en compte la necessitat de l'emmagatzematge; i en cas afirmatiu ell mateix se'n fa responsable.

***Creador orb* en termes de Cohesió.** La coincidència entre *creador* i *emmagatzemador* dona més *cohesió* al component que té ambdues responsabilitats, enfront de la coincidència entre *creador* i *usuari*. En el primer cas el component gestiona el *cicle de vida* de l'objectiu (una sola responsabilitat, tot i que amb diferents tasques: creació, emmagatzematge, destrucció,¹⁰ etc). En el segon cas el component gestiona part del *cicle de vida* de l'objectiu (la creació, però no pas l'emmagatzematge) i la utilització que es fa de l'objectiu; el resultat són com a mínim dues responsabilitats diferents.

¹⁰Que algú sigui el creador d'un objecte no significa necessàriament que en sigui també el destructor. Però és una situació força comuna. Per tant, essent primmirats, caldria dir que la coincidència entre *creador* i *emmagatzemador* concentra en un únic component diferents etapes del *cicle de vida*, tot i que no necessàriament totes.

4 Emmagatzematge

4.1 Estudi del MC

? (18)

Qui és el responsable d'emmagatzemar una nova realització?

Visibilitat d'atribut: un mecanisme amb dues funcions. La visibilitat d'atribut és el mecanisme per modelitzar les associacions del MC. Però també és el mecanisme emprat per l'emmagatzematge.

(24)

La *visibilitat d'atribut*:

- Modelitza una associació de MC
- Expressa la potencialitat de col·laboració o comunicació que és independent del context d'emissió
- És el mecanisme emprat per l'emmagatzematge d'objectes

Associacions i emmagatzematge. La visibilitat d'atribut fa coincidir en un sol mecanisme dues necessitats diferents: la modelització d'associacions i l'emmagatzematge d'objectes. Sempre que sigui possible interessarà mantenir com a visibilitats d'atribut aquelles que responguin a ambdues finalitats.¹¹→ D'aquí que l'estudi del MC ens dona pautes per l'assignació de la responsabilitat d'emmagatzematge.

10.4.5.Principi de l'Emmagatzemador canònic,
pàgina 172

Estabilitat de l'especificació. La dificultat de detectar els usuaris futurs i la inestabilitat del qualificatiu "present" o "futur" sobre un usuari contrasta amb l'estabilitat de les associacions de MC. L'especificació pot ser errònia o modificable, i és possible que durant el disseny ens n'adonem que cal fer canvis en l'especificació (per incomplerta, ambigua o inconsistent); sigui com sigui, però, hom pretén l'*estabilitat de l'especificació*, per ser aquesta el contracte del disseny.

Inestabilitat del disseny. Els principis de la *Teranyina* i de la *Reconsideració* expressen la *inestabilitat del disseny* davant d'una mateixa especificació: la consistència entre diferents decisions i objectius pot obtenir-se diferents maneres (*Teranyina*); la incrementalitat del desenvolupament dóna inestabilitat a les decisions preses (*Reconsideració*).

¹¹En l'apartat 10.4.5.Principi de l'Emmagatzemador canònic, pàgina 172, estudiem què passa en cas de no fer-hi així.

Eficàcia de l'estudi de MC. L'estabilitat de l'especificació respecte la inestabilitat (almenys durant el desenvolupament) del disseny, ens diu que si volem construir un principi general per assignar la responsabilitat de creació d'una nova realització llavors l'estudi de MC és una bona idea (ja que aquest, com que forma part de l'especificació, és estable).

4.2 Visibilitat d'emmagatzematge

Context de l'anàlisi. Suposem que volem crear una realització de l'objectiu $o : O$. Sigui una associació $M - N$ entre els conceptes C i O ,¹² i sigui $c : C \rightarrow o : O$ un enllaç d'aquesta associació.

Visibilitat $O \xrightarrow{atr} C^*$. La visibilitat d'atribut $O \xrightarrow{atr} C^*$ com a model l'associació \rightarrow no té sentit de cara a l'emmagatzematge de l'objectiu: ningú no l'emmagatzema; si de cas és ell qui emmagatzema $c : C$.

3.5. Modelització de les associacions del MC,
pàgina 75

Visibilitat $C \xrightarrow{atr} O^*$. Per contra, la visibilitat $C \xrightarrow{atr} O^*$ sí que és útil de cara a l'emmagatzematge de l'objectiu: $c : C$ l'emmagatzema (a ell, i possiblement també a d'altres objectes; d'aquí la visibilitat generalitzada).

La modelització d'una associació és una visibilitat d'atribut orientada segons les necessitats de l'emmagatzematge: de l'emmagatzemador cap a l'objectiu.

(25)

4.3 Principi de l'Emmagatzemador selectiu

Emmagatzemador selectiu i Espill. El principi *Emmagatzemador selectiu* \rightarrow , que veurem més endavant, el que pretenen és evitar que l'emmagatzematge violi el principi de l'*Espill*: les visibilitats d'atribut que proposarà tenen la doble funcionalitat de modelitzar una associació i de servir de mecanisme d'emmagatzematge. Ho veiem tot seguit.

Principi, pàgina 168

Context de l'anàlisi. En el que segueix usarem la següent terminologia:

- Sigui un objecte $o : O$, anomenat **objectiu**
- Sigui un objecte $c : C$, anomenat **objecte emmagatzemador**
- Siguin $o : O$ i $c : C$ dues entitats tals que $model(o : O) = o : O$ i $model(c : C) = c : C$ ¹³ Anomenen a aquestes entitats **origen de l'objectiu** i **origen de l'emmagatzemador**, respectivament.

¹²Deixem el cas de les associacions no binàries per més endavant.

¹³La distinció entre una entitat i l'objecte que n'és el seu model s'obté pel context i per la tipografia: *objecte* versus *entitat*.

- Sigui un enllaç dirigit $c : C \rightarrow O^*$ tal que l'objectiu $o : O$ pertany a l'abast de $c : C \rightarrow O^*$; és a dir, l'objectiu és una de les destinacions de l'enllaç dirigit $c : C \rightarrow O^*$. Anomenem **enllaç dirigit d'emmagatzematge** aquest enllaç dirigit.

$$o \in \text{abast}(c : C \rightarrow O^*)$$

- Sigui la visibilitat $C \rightarrow O^*$ aquella de la qual l'enllaç dirigit d'emmagatzematge n'és una realització. Anomenem **visibilitat d'emmagatzematge** aquesta visibilitat.
- Sigui l'associació $C - O$ tal que el seu model és la visibilitat d'emmagatzematge. Anomenem **origen de la visibilitat d'emmagatzematge** aquesta associació.
- Sigui l'enllaç $c : C - o : O$ realització de l'origen de l'associació d'emmagatzematge entre l'origen de l'emmagatzemador i l'origen de l'objectiu. Anomenem **origen de l'enllaç dirigit d'emmagatzematge** aquest enllaç.

Primeres observacions. Si analitzem detingudament el context de l'anàlisi acabat de plantejar ens adonem que per tal que l'emmagatzematge no violi *Espill* cal assegurar dues condicions:

- **A nivell de visibilitats**

Donada una visibilitat d'emmagatzematge, en el MC considerat ha d'existir l'associació que és l'origen d'aquesta visibilitat

- **A nivell d'enllaços dirigits**

Donat un enllaç dirigit d'emmagatzematge, en el MC considerat ha d'existir el corresponent origen d'aquest enllaç dirigit

No violació d'*Espill* a nivell de les visibilitats. Per assegurar que una visibilitat d'emmagatzematge no viola *Espill*, cal assegurar que les úniques visibilitats d'emmagatzematge permeses són aquelles que són el model d'alguna associació del MC. D'aquesta manera assurem l'existència de l'associació *origen* de qualsevol visibilitat d'emmagatzematge.

No violació d'*Espill* a nivell dels enllaços dirigits. Per no violar *Espill* a nivell dels enllaços dirigits, cal assegurar que l'enllaç origen de l'enllaç dirigit d'emmagatzematge és un enllaç existent. Això significa:

1. Existeix l'associació origen de la visibilitat d'emmagatzematge
2. Existeix un enllaç, realització de l'associació origen de la visibilitat d'emmagatzematge, que el seu model és l'enllaç dirigit d'emmagatzematge considerat

Una condició repetida. De les dues condicions exigides per evitar la violació d'*Espill* a nivell dels enllaços dirigits, la primera (a saber, l'existència de l'associació origen de la visibilitat d'emmagatzematge) és la mateixa condició exigida per evitar la violació d'*Espill* a nivell de les visibilitats.

Una condició que cal assegurar. Per tant, a nivell dels enllaços dirigits només ens hem de preocupar de l'existència de l'enllaç que és l'origen de l'enllaç dirigit d'emmagatzematge.

El camí a la inversa. Enlloc de proposar un objecte emmagatzemador, i a continuació comprovar que l'enllaç format pel seu origen i l'origen de l'objectiu és un enllaç existent, ho farem a la inversa, a l'igual com hem fet amb les visibilitats.

Anàlisi que farem. Partirem dels enllaços que poden ser origen d'un enllaç d'emmagatzematge, i com a emmagatzemador prendrem l'objecte que sigui el model de l'extrem que no estigui modelitzat per l'objecte objectiu. Tot seguit ho veiem detingudament.

Anàlisi pas a pas. Anem a seguir el procediment pas a pas per trobar un emmagatzemador per a un objectiu $\circ : \circ$ qualsevol:

- Sigui un objectiu $\circ : \circ$
- Per *Espill* existeix una entitat $o : O$ tal que $model(o : O) = \circ : \circ$
- La condició sobre les visibilitats assegura que:
 - Tenim una visibilitat d'emmagatzematge $C \rightarrow \circ^*$
 - Tenim una associació $C - O$ modelitzada per la visibilitat d'emmagatzematge
- La qüestió ara és quin objecte $c : C$ pot ser l'emmagatzemador de $\circ : \circ$
- *Hipòtesi.* Suposem que existeix una entitat $c : C$ tal que existeix l'enllaç $c : C - o : O$, realització de la visibilitat $C - O$
- Per la completesa del model,¹⁴ l'existència de l'entitat $c : C$ assegura l'existència d'un objecte $c : C$ tal que $model(c : C) = c : C$
- Per la completesa del model,¹⁵ l'existència de l'enllaç $c : C - o : O$, assegura l'existència d'un enllaç dirigit $c : C \rightarrow \circ^*$ que el modelitza; això és, un enllaç dirigit tal que $\circ : \circ$ és una de les seves destinacions

¹⁴No entrarem aquí en què significa exactament la completesa del disseny, anàlisi que farem més endavant. \rightarrow Pel que ens interessa en aquests moments n'hi ha prou en dir que un model és complert quan, entre d'altres coses, si introdueix un component que és el model d'un concepte, llavors totes les entitats realització d'aquest concepte també es modelitzen, i en concret ho fan com a objectes que són realització del component considerat.

¹⁵No entrarem aquí en què significa exactament la completesa del disseny, anàlisi que farem més endavant. \rightarrow Pel que ens interessa en aquests moments n'hi ha prou en dir que un model és complert quan, entre d'altres coses, si introdueix una visibilitat (d'atribut) que és el model d'una associació, llavors tots els enllaços realització d'aquesta associació també es modelitzen, i en concret ho fan com a enllaços dirigits que són realització de la visibilitat considerada.

- Per tant, podem usar $c : C \rightarrow o : O$ com a enllaç dirigit d'emmagatzematge; en d'altres paraules, podem usar $c : C$ com a emmagatzemador de $o : O$

Conclusions de l'anàlisi. De l'anàlisi pas a pas que acabem de fer en podem concloure que, un cop assegurada l'existència de l'associació que és origen de la visibilitat d'emmagatzematge, n'hi ha prou en exigir l'existència d'un enllaç, realització d'aquesta associació, en el que hi participin les entitats modelitzades pels objectes objectiu i creador.

Principi



Emmagatzemador selectiu. La responsabilitat d'emmagatzemar les realitzacions d'un component O :¹⁶

1. L'assignem a un *component* C tal que en el MC hi ha una *associació* entre O i C
2. L'executem en un *objecte* $c : C$ tal que en l'associació $C - O$ considerada, hi ha un *enllaç* $c : C - o : O$

Emmagatzemador selectiu a nivell de les abstraccions. A nivell de les abstraccions el principi de l'*Emmagatzemador selectiu* diu que només les visibilitats que són el model d'una associació poden ser visibilitat d'emmagatzematge. Per tant el MC ens dona pistes dels possibles emmagatzemadors.

Emmagatzemador selectiu a nivell de les realitzacions. A nivell de les realitzacions el principi de l'*Emmagatzemador selectiu* diu que el component escollit com a emmagatzemador només podrà exercir la seva responsabilitat si en el MC hi ha un enllaç en el que participi l'entitat modelitzada per l'objectiu $o : O$.

Emmagatzemador selectiu a nivell de les realitzacions i el MC. A nivell de les realitzacions el principi de l'*Emmagatzemador selectiu* exigeix la presència d'un enllaç a MC. És a dir, exigeix que per tal de poder emmagatzemar, l'emmagatzemador i l'objectiu emmagatzemat han de ser el model d'entitats semànticament lligades a MC. El problema és que en el DMC no hi apareixen les realitzacions i per tant, gràficament no podem detectar aquests lligams semàntics

Què aconseguim amb aquestes dues condicions. Amb les dues condicions que imposa l'*Emmagatzemador selectiu* assegurem:

- **Condició sobre les abstraccions.** Assegura que la visibilitat d'emmagatzematge emprada és el model d'una associació del MC, i per tant no viola *Espill* (a nivell de les abstraccions).¹⁷

¹⁶Assumim $model(C) = C$, $model(O) = O$, $model(c:C) = c : C$ i $model(o:O) = o : O$.

¹⁷Més endavant \rightarrow veurem les condicions que ha de tenir un model per estar ben construït. I

Demostració.

Sigui C un *emmagatzemador* de \circ i suposem que a MC no hi ha cap associació entre C i O . L'emmagatzematge significa l'existència d'una visibilitat d'atribut $C \xrightarrow{atr} O^*$; i aquesta visibilitat és interpretable en termes de model d'una associació entre C i O . Per tant tenim un element de MComp que modelitza un element inexistent a MC: hem violat *Espill*. \square

- **Condicció sobre les realitzacions.** Assegura que els enllaços dirigits d'emmagatzematge que tinguem són el model d'enllaços presents en el MC. Per tant els enllaços dirigits d'emmagatzematge no violen *Espill*.

Demostració.

La demostració és similar a la condició sobre les abstraccions: tot enllaç dirigit que emmagatzema un objectiu $\circ : \circ$ es pot interpretar com a model d'un enllaç. Si no volem violar *Espill* l'emmagatzematge ha de coincidir amb els enllaços. \square

El perquè de l'adjectiu *selectiu*. El principi *Emmagatzemador selectiu* assigna la responsabilitat de la creació dels objectes realització de \circ a un component C . Però la condició sobre els enllaços que imposa el que diu és que no tot objecte $\circ : \circ$ tindrà un emmagatzemador $c : C$. D'aquí que parlem de *selectiu*: si entenem les associacions com una funció, i suposem l'associació $f :: C \rightarrow O$ com l'origen de la visibilitat d'emmagatzematge, llavors només podrem exercir la responsabilitat de l'emmagatzematge sobre aquells objectes $\circ : \circ$ tals que el seu origen pertanyi a l'abast de f ; o dit altrament, només els objectes que són el model de les entitats de l'abast de f poden ser emmagatzemats.

Definició. Emmagatzemador selectiu

Els emmagatzemadors proposats pel principi Emmagatzemador selectiu reben el nom d'emmagatzemadors selectius

4.4 Principi de l'Emmagatzemador general

Notació. En aquest apartat seguim la nomenclatura introduïda en la pàgina 165. \rightarrow

Emmagatzemador selectiu en termes planers. El principi de l'emmagatzemador selectiu \rightarrow proposa un component com a responsable de l'emmagatzematge, però alhora posa unes limitacions sobre quins objectes podran ser realment emmagatzemats en l'exercici de la responsabilitat assignada.

arribarem a la conclusió que només ens hem de preocupar, en modelitzar, de les abstraccions: assumim que si una abstracció és el model d'una altra, les seves realitzacions són el model de realitzacions de l'abstracció que estem modelitzant. Per tant, la condició sobre les realitzacions de l'Emmagatzemador selectiu queda subsumida.



Els objectes seleccionats. Per tal que un objecte pugui ser emmagatzemat per la visibilitat d'emmagatzematge proposada pel principi de l'*Emmagatzemador selectiu*, cal que sigui el model d'alguna entitat que participa en algun enllaç de l'associació que és l'origen de la visibilitat d'emmagatzematge.

? (19)

Un cop assignada la responsabilitat de l'emmagatzematge a una visibilitat $C \rightarrow O^*$, com assegurem que tot objectiu $o : O$ és realment emmagatzemable?

Què ens preguntem exactament. La pregunta acabada de formular el que planteja és com assegurem que donat un enllaç dirigit d'emmagatzematge qualsevol podem assegurar que l'origen d'aquest enllaç dirigit és un enllaç present en el MC (present, i no pas potencialment possible, cosa que sabem certa).

Universalitat de la pregunta. És important observar com el que ens preguntem és com assegurem l'existència de l'emmagatzemador donat un objectiu qualsevol. Per tant la resposta no pot dependre d'un objectiu concret; sinó que ha de ser una resposta universal.

Principi Emmagatzemador selectiu, pàgina 168

Aplicació de l'Emmagatzemador selectiu. Sabem \rightarrow que si $o : O$ és emmagatzemable, llavors ha d'existir un enllaç $c : C - o : O$, per alguna entitat $c : C$, tal que sigui realització de l'associació que modelitzem amb la visibilitat d'emmagatzematge, i on $model(o : O) = o : O$.

Pregunta reformulada. L'aplicació de l'*Emmagatzemador selectiu* permet reformular la pregunta que estem intentar resoldre:

? (20)

Donat un objecte $o : O$ qualsevol, amb $model(o : O) = o : O$, com assegurarem que sempre hi ha una entitat $c : C$ tal que $c : C - o : O$ és un enllaç realització d'una associació $C - O$

Resposta simple. La pregunta acabada de formular té una resposta ben simple: n'hi ha prou que l'associació $C - O$ considerada sigui obligatòria a l'extrem de C .

Demostració.

- *Espill* assegura que donat l'objecte $o : O$ existeix l'entitat $o : O$ tal que $model(o : O) = o : O$
- L'obligatorietat de l'associació assegura que donada una entitat $o : O$ sempre hi ha un enllaç $c : C - o : O$, realització de l'associació $C - O$
- L'existència de l'enllaç assegura l'existència dels seus extrems; en concret, l'existència de l'entitat $c : C$

- La completesa¹⁸ del model assegura que hi ha un objecte $c:C$ tal que $model(c:C) = c:C$

□

Simplificació de les condicions. La condició de l'obligatorietat de l'associació $C - O$ per l'extrem de C assegura l'existència dels enllaços que són origen dels enllaços dirigits d'emmagatzematge. Això permet simplificar les condicions exigides per a l'emmagatzematge:

Emmagatzemador general. La responsabilitat d'emmagatzemar les realitzacions d'un component O l'assignem a un component C tal que¹⁹ en el MC hi ha una associació entre O i C , **obligada per l'extrem de C** .

Principi



Principi en termes de components. El principi *Emmagatzemador selectiu* és un principi estrany, en el sentit que està expressat en termes d'abstraccions i de realitzacions. Per contra, el principi de l'*Emmagatzematge general* només està expressat en termes d'abstraccions.

Una condició innecesària. L' *Emmagatzemador selectiu* ens diu que no tot objectiu té un enllaç dirigit d'emmagatzematge correcte. El principi de l'*Emmagatzemador general* no imposa cap condició addicional perquè sota les seves condicions per a tot objectiu hi ha un enllaç dirigit d'emmagatzematge correcte.

Una condició implícita. Tant en un principi com l'altre, però, els enllaços dirigits introduïts no poden violar *Espill*. En el cas de *Emmagatzemador selectiu* aquesta condició l'explicitem; en el cas de l'*Emmagatzemador general* la condició no l'explicitem perquè entenem que queda subsumida pel principi de l'*Espill*, que sempre volem complir.²⁰

Definició. Emmagatzemador general

Els emmagatzemadors proposats pel principi Emmagatzemador general reben el nom d'emmagatzemadors generals

¹⁸No entrarem aquí en què significa exactament la completesa del disseny, anàlisi que farem més endavant. → Pel que ens interessa en aquests moments n'hi ha prou en dir que un model és complert quan, entre d'altres coses, si introdueix un component que és el model d'un concepte, llavors totes les entitats realització d'aquest concepte també es modelitzen, i en concret ho fan com a objectes que són realització del component considerat.

¹⁹Assumim $model(C) = C$ i $model(O) = O$.

²⁰En el cas de l'*Emmagatzemador selectiu* imposen la condició que existeixi l'origen de l'enllaç dirigit d'emmagatzematge; i per expressar aquesta condició explicitem que l'enllaç dirigit d'emmagatzematge és el model d'un enllaç dirigit de MC. En el cas de l'*Emmagatzemador general* les condicions sobre la visibilitat asseguren l'existència dels enllaços necessaris.

4.5 Principi de l'Emmagatzemador canònic

L'Emmagatzemador general no és massa precís. Suposem que l'Emmagatzemador general proposa que el creador de C és O . Suposem un objectiu $o : O$ qual-sevol. El principi de l'Emmagatzemador general assegura que hi ha un possible emmagatzemador $c : C$ dins de la visibilitat d'emmagatzematge seleccionada. Però i si n'hi ha més d'un?

Suprimim la indefinició. Com acabem de veure, el principi l'Emmagatzemador general pot proporcionar més d'un objecte creador per a un mateix objectiu. La monoavaluació que exigeix el principi de l'Emmagatzemador canònic evita aquesta situació.

Principi



Emmagatzemador (canònic).

La responsabilitat d'emmagatzemar les realitzacions d'un component O l'assignem a un component C tal que²¹ hi ha una associació entre O i C en el MC, *obligada i monoavaluada* per l'extrem de C

Definició. Emmagatzemador canònic

Els emmagatzemadors proposats pel principi Emmagatzemador reben el nom d'emmagatzemadors canònics



Notació. Sovint per referir-nos al principi de l'Emmagatzemador canònic simplement parlarem del principi de l'Emmagatzemador

Emmagatzemador general o canònic. L'anàlisi de quan cal usar un principi o un altre la deixem pel proper apartat,[→] on analitzem el lligam entre l'emmagatzematge i la creació.

²¹Assumim $model(C) = C$ i $model(O) = O$.

5 Principi del *Creador canònic*

5.1 Creació i emmagatzematge

Exigències per a la creació. El principi del *Creador orb*, \rightarrow demana que el creador d'un objectiu també en sigui el seu emmagatzemador.

Principi, pàgina 163

Exigències per a l'emmagatzematge. Els principis d'emmagatzematge acabats de veure, \rightarrow imposen condicions sobre qui pot ser emmagatzemador o no. El que farem a continuació és aplicar aquests principis en el cas d'una creació.

10.4.1. Estudi del MC, pàgina 164

Context de l'anàlisi. Suposem que volem crear un objecte $o : O$. Tots els tres principis d'emmagatzematge vistos exigeixen una visibilitat d'emmagatzematge $C \xrightarrow{atr} O^*$ que és el model d'una associació $C - O$ present al MC.

Emmagatzemador selectiu. Suposem que per escollir l'emmagatzemador exigit pel *Creador orb* usem el principi *Creador selectiu*: \rightarrow

Principi, pàgina 168

- Com hem vist, \rightarrow hi ha alguns objectes $o : O$ que no tenen emmagatzemador
- Com a conseqüència hi haurà alguns objectes que no tindran creador

10.4.3. Principi de l'Emmagatzemador selectiu, pàgina 165

Desig d'universalitat. El principi de creació que estem cercant és un que ens digui, donat un objecte $o : O$ qualsevol, quin o quins objectes en poden ser els seus creadors. Com acabem de veure l'*Emmagatzemador selectiu* no és útil amb aquests propòsits.

Emmagatzemador general. Suposem que per escollir l'emmagatzemador exigit pel *Creador orb* usem el principi *Emmagatzemador general*: \rightarrow :

Principi, pàgina 171

- Amb l'aplicació de l'*Emmagatzemador general* aconseguim la universalitat cercada: tot objecte té el seu creador
- Què passa però si, escollida la visibilitat d'emmagatzematge, hi ha més d'un candidat a creador? No tindrem més remei que escollir entre un dels diversos objectes candidats

Problemes de decidir. Suposem que el principi de l'*Emmagatzemador general* proposa una visibilitat d'emmagatzematge $C \xrightarrow{atr} O^*$, i que per a un determinat objecte $o : O$ proposa dos possibles enllaços dirigits d'emmagatzematge: $c_1 : C \rightarrow i^*$ $c_2 : C \rightarrow .^*$ I suposem que, pel motiu que sigui, decidim que l'emmagatzemador que requereix el *Creador orb* és l'objecte c_1 . Aquesta decisió comporta els següents problemes:

- **Necessitat de recordar.** Hem decidit que el creador de $o : o$ és c_1 , que el crea i l'emmagatzema. Més tard, quan vulguem recuperar l'objecte $o : o$ haurem de recordar quina decisió hem pres. Si analitzem el MC, segons els criteris del principi de l'*Emmagatzemador general*, veurem que hi ha dos possibles creadors i emmagatzemadors: c_1 i c_2 . Però quin dels dos vam escollir en el moment de dissenyar la creació de $o : o$?
 - La necessitat de recordar no és una dificultat insalvable. Complica el desenvolupament, però no entra en contradicció amb cap principi
- **Guerra fratricida.** Aquest és un problema molt més greu que l'anterior. Analitzem-ho.
 - L'elecció de c_1 com a creador de $o : o$, impossibilita que ho sigui c_2 (o qualsevol altre): un objecte pot tenir diversos emmagatzemadors, però només un creador
 - Si escollim c_1 com a creador de $o : o$, és perquè aquesta és una potencialitat prevista en el component o .
 - En concret, sabem que qualsevol enllaç dirigit $c : C \rightarrow O^*$, realització de la visibilitat d'emmagatzematge $C \rightarrow O^*$, i que contigui l'objecte $o : o$ en les seves destinacions, és un possible enllaç dirigit d'emmagatzematge, sempre i quan existeixi l'enllaç $c : C - o : O$, on $model(c:C) = c:C$ i $model(o:O) = o:o$
 - En el nostre exemple tant c_1 com c_2 tenen la potencialitat de ser creadors. Però en el moment que la donem a c_1 , l'objecte c_2 la perd

Exemple 56 (Emmagatzemador general) Els empleats d'una empresa s'assignen a departaments. Tenim una associació `assig:Empleat-Departament`; amb multiplicitats $M - N$, ja que un mateix empleat pot fer feines a més d'un departament.

Suposem que ens decidim per `Departament` com a creadora d'`Empleat`. És a dir, tenim la visibilitat d'emmagatzematge $Departament \xrightarrow{atr} Empleat^*$. Llavors `Departament` és un emmagatzemador general d'`Empleat`. (Per veure-ho n'hi ha prou en observar que `assig` és obligada per l'extrem `Departament`).

En Joan està assignat a `Desenvolupament` i a `Màrqueting`. Quin dels dos departaments crea l'empleat Joan? Si el crea `Desenvolupament` estem dient que el lligam d'en Joan amb aquest departament és molt més fort que el que té amb `Màrqueting`. De fet segurament pensarem que en Joan està assignat a `Desenvolupament`, però que aquest departament el cedeix a estones a `Màrqueting`.

La darrera observació permet pensar en dues associacions entre `Departament` i `Empleat`: pertinença,

$1 - N$; i col·laboració, $M - N$. Així les coses, la visibilitat d'emmagatzematge és el model de pertinença; i per tant l'emmagatzemador és canònic (és el model d'una associació, que és obligada i monoavaluada per l'extrem de Departament).

Conclusió. En general la necessitat d'entrar en una guerra fratricida és un símptoma de tenir un mal model. En l'exemple, enlloc d'assignar és preferible tenir pertinença i col·laboració. Per tant l'ús d'emmagatzemadors generals és conseqüència d'una mala especificació.

- **Manca de simetria.** Com a conseqüència de la guerra fratricida, hi ha una pèrdua important de simetria. Dins la visibilitat d'emmagatzematge $C \rightarrow O^*$ tots els enllaços dirigits són iguals. Però l'elecció de c_1 com a creador de $o : o$ trenca aquesta igualtat: l'enllaç dirigit amb origen c_1 i que conté $o : o$ com a destinació és un enllaç dirigit obligatori (es crea en el moment de crear $o : o$), mentre que l'enllaç dirigit amb origen c_2 i que conté $o : o$ com a destinació és un enllaç dirigit optatiu

Creació i emmagatzematge.

Per tal de determinar quin és l'emmagatzemador exigit pel *Creador orb*, ni el principi *Emmagatzemador selectiu* ni el principi *Emmagatzemador general* són vàlids

(26)

Emmagatzemador canònic. Suposem que per escollir l'emmagatzemador exigit pel *Creador orb* usem el principi *Emmagatzemador canònic*: \rightarrow Aquest principi assegura que tot objectiu $o : c$ té un emmagatzemador, i que aquest és únic. Per tant no tenim cap dels problemes que teníem amb l'*Emmagatzemador general*.

Principi, pàgina 172

A punt de solfa. Ara ja tenim tot el coneixement necessari per poder presentar el *principi del Creador canònic*.

5.2 Creador canònic

Creador canònic. *Presentació en termes dels creadors.*²² Els candidats a creador són els experts, si cal per delegació, de la creació que a més són emmagatzemadors canònics.

En el cas de creació d'ús (creació sense emmagatzematge) els candidats a creador són els experts, si cal per delegació, de la creació que a més en són usuaris.

Principi



Definició. Creador canònic

Els candidats a creador proposats pel principi Creador canònic reben el nom de creadors canònics.

Subsumpció d'altres principis. El principi del *Creador canònic* en essència té les mateixes motivacions i propòsits que el principi del *Creador orb*, però es restringeix als *emmagatzemadors canònics*.

Emmagatzematge consistent amb MC. La limitació dels *emmagatzemadors* als proposats pel principi *Emmagatzemador* (canònic), és a dir, als *emmagatzemadors canònics*, assegura que la necessitat d'emmagatzematge no creï lligams que violin el principi de la *Modelització contínua*.

Estabilitat de la proposta. L'obligació de fer coincidir el *creador* amb algun *emmagatzemador* evita la problemàtica de detectar un conjunt estable d'*usuaris*.

6 Emmagatzematge primari

6.1 Creador canònic i emmagatzematge primari

Definició. Emmagatzematge primari

Anomenem Emmagatzematge primari a tota visibilitat d'atribut tal que la seva hipotètica inversa sigui obligada i monoavaluada

Inversa hipotètica. Les visibilitats tenen una orientació. La definició d'*Emmagatzematge primari* el que demana és que considerem la visibilitat en sentit contrari. Aquesta visibilitat inversa pot ser que existeixi o no, d'aquí que diguem que és hipotètica.

Excursió. (Visibilitat inversa) Sigui una visibilitat $A \rightarrow B^*$. Sigui un enllaç dirigit $a : A \rightarrow B^*$, realització d'aquesta visibilitat. I sigui $b : B$ un objecte que pertany a la destinació de l'enllaç dirigit considerat. Llavors, la visibilitat inversa de $A \rightarrow B^*$ té un enllaç dirigit $b : B \rightarrow A^*$ tal que $a : A$ pertany a la destinació de $b : B \rightarrow A^*$. Formalment:

$$o : O \in \text{abast}(c : C \rightarrow O^*) \Leftrightarrow c : C \in \text{abast}(o : O \rightarrow C^*)$$

²²Més endavant \rightarrow donem una presentació alternativa d'aquest principi, que enlloc de fixar-se en qui és el creador (component creador), es fixa en on s'emmagatzema (visibilitat d'emmagatzematge). Ambdues presentacions són gairebé equivalents ja que el creador canònic exigeix que el creador també sigui emmagatzemador, és a dir, que sigui l'origen d'un enllaç dirigit d'emmagatzematge. La diferència entre ambdues presentacions és que quan emfasitzem en el creador (la presentació actual) tenim en compte totes les creacions possibles, àdhuc si no hi ha emmagatzematge. En canvi, quan emfasitzem en com s'emmagatzema (la presentació que veurem més endavant), no considerem el cas de la creació sense emmagatzematge.

Emmagatzematge primari i emmagatzemador canònic. El principi de l'*Emmagatzemador canònic* proposa un objecte creador a partir de l'anàlisi de les associacions de MC. La visibilitat d'emmagatzematge introduïda per un emmagatzemador canònic és un **emmagatzematge primari**.

Demostració.

- Sigui $c : C$ l'emmagatzemador canònic de $o : O$. Per tant tenim la següent visibilitat d'emmagatzematge: $C \xrightarrow{atr} O^*$
- Per *Emmagatzemador canònic*, $\rightarrow model(C \rightarrow O^*) = C - O$ on $C - O$ és una associació de MC
- Per *Emmagatzemador canònic*, \rightarrow l'associació $C - A$ és obligada i monoavaluada per l'extrem de C
- Per tant, donada una entitat $o : O$ qualsevol sempre hi ha una entitat $c : C$ tal que $c : C - o : O$ és un enllaç realització de l'associació $C - O$. I aquesta entitat $c : C$ és única.
- Suposem que volem modelitzar l'associació $C - O$ amb la visibilitat $O \rightarrow C^*$.²³ Per completesa²⁴ del model, l'enllaç $c : C - o : O$ ha d'aparèixer en el model en termes d'un enllaç dirigit $o : O \rightarrow c : C^*$
- Com que donada una entitat $o : O$ només hi ha un únic enllaç $c : C - o : O$, el seu model $o : O \rightarrow c : C^*$ ha de ser monoavaluat (i obligat)
- Per tant, la visibilitat que modelitza $C - O$ és $O \rightarrow C$ (monoavaluada i amb abast tot C).
- La visibilitat $O \rightarrow C$ és la inversa de la visibilitat d'emmagatzematge, $C \rightarrow O^*$

□

Presentació alternativa del Creador canònic. A partir del resultat anterior podem expressar el principi del creador canònic de la següent manera:

²³Compte: no estem usant la visibilitat d'emmagatzematge, ans la seva inversa.

²⁴No entrarem aquí en què significa exactament la completesa del disseny, anàlisi que farem més endavant. \rightarrow Pel que ens interessa en aquests moments n'hi ha prou en dir que un model és complert quan, entre d'altres coses, si introdueix una visibilitat (d'atribut) que és el model d'una associació, llavors tots els enllaços realització d'aquesta associació també es modelitzen, i en concret ho fan com a enllaços dirigits que són realització de la visibilitat considerada.



Creador canònic. Presentació en termes de l'emmagatzematge.²⁵ Les visibilitats d'emmagatzematge conseqüència de la creació han de ser *emmagatzematges primaris* que no violin *Espill*.

Emmagatzematge i creació. La creació significa emmagatzematge; ja ens ho deia el principi *Creador orb*. El principi del *Creador canònic* ens diu que aquest emmagatzematge, però, no pot ser qualsevol emmagatzematge: ha de ser un *emmagatzematge primari*. Els detalls els hem vist a, 10.5.1.Creació i emmagatzematge[→] tot i que allà no usàvem aquesta terminologia:

- Tothom té un creador. Això és, la hipotètica inversa de la visibilitat d'emmagatzematge emprada en la creació ha de ser obligada.
- Un cop escollit el component creador, tothom té com màxim un objecte creador. Això és, la hipotètica inversa de la visibilitat d'emmagatzematge emprada en la creació ha de ser mono-avaluada.

Creador canònic i Espill. En l'anàlisi de la creació i l'emmagatzematge[→] que ens ha dut al *Creador canònic*, el desig de no violar *Espill* ens ha portat a exigir que l'*emmagatzematge primari* escollit no violi *Espill*. I això és el que diu la reformulació del *Creador canònic* que acabem de presentar.

6.2 Anàlisi del *Creador orb*

6.2.1 Tipificació dels creadors orbs

Camí recorregut. Des de la introducció del principi del *Creador orb* [→] hem avançat molt en la comprensió de què significa l'emmagatzematge. En l'anàlisi de l'emmagatzematge exigint per la creació[→] hem vist que ens interessa un emmagatzemador canònic. Justament el principi del *Creador canònic* [→] diu que com a creador cal usar un emmagatzemador canònic; o, equivalentment, que la visibilitat d'emmagatzematge necessària per a la creació és un emmagatzematge primari.[→]

Tipologia de creadors orbs. Amb tot aquest coneixement anem a classificar els diferents tipus de *creador orb*. La tipologia es basa en les propietats de la *visibilitat d'emmagatzematge* emprada per la creació; el creador considerat és l'origen d'aquesta visibilitat

²⁵ Aquí donem donem una presentació alternativa del principi del *Creador canònic*,[→] que enlloc de fixar-se en qui és el creador (component creador), es fixa en on s'emmagatzema (visibilitat d'emmagatzematge). Ambdues presentacions són gairebé equivalents ja que el creador canònic exigeix que el creador també sigui emmagatzemador, és a dir, que sigui l'origen d'un enllaç dirigit d'emmagatzematge. La diferència entre ambdues presentacions és que quan emfasitzem en el creador (la presentació anterior) tenim en compte totes les creacions possibles, àdhuc si no hi ha emmagatzematge. En canvi, quan emfasitzem en com s'emmagatzema (la presentació actual), no considerem el cas de la creació sense emmagatzematge.

- **Creador canònic.** La visibilitat d'emmagatzematge és el model d'una associació de MC;²⁶ la seva hipotètica inversa és obligada i monoavaluada. Alternativament, *la visibilitat d'emmagatzematge és un emmagatzematge primari que no viola Espill*
 - Tot creador orb és l'origen d'una visibilitat d'emmagatzematge. El principi de l'*Creador canònic* exigeix que aquesta visibilitat sigui un *emmagatzemador primari* i que no violi *Espill*
 - Per tant, tot creador canònic també és un creador orb.
- **Creador canònic fals.** La visibilitat d'emmagatzematge és el model d'una associació de MC qualsevol.²⁷ Per poder analitzar el contrast amb el *Creador canònic*, direm que un *Creador canònic fals* és aquell en què *la visibilitat d'emmagatzematge no viola Espill*. (Fixem-nos que hem perdut la restricció de ser emmagatzematge primari).
 - **Emmagatzemador selectiu.** Enlloc de considerar els emmagatzemadors canònics considerem com a emmagatzemadors els *emmagatzemadors selectius* →
 - **Emmagatzemador general.** Enlloc de considerar els emmagatzemadors canònics considerem com a emmagatzemadors els *emmagatzemadors generals* →
- **Creador orb pur.** *La visibilitat d'emmagatzematge no és el model de cap associació de MC*
 - Donat un *creador canònic* la visibilitat d'emmagatzematge té dues funcions: modelitzar una associació i mantenir l'emmagatzematge.
 - Donat un *creador orb pur* la visibilitat d'emmagatzematge només té la funció de mantenir l'emmagatzematge.
- **Creador orb enganyós.** Els enllaços dirigits d'emmagatzematge són realitzacions d'una visibilitat d'emmagatzematge que és el model d'una associació; però els enllaços dirigits d'emmagatzematge no necessàriament han de ser el model d'enllaços presents a MC. És a dir, *la visibilitat d'emmagatzematge no viola Espill, però els enllaços dirigits d'emmagatzematge sí poden violar-lo*.
 - *Creador canònic* i *Creador canònic fals* no violen *Espill*; *Creador orb pur* el viola a nivell de les abstraccions; *Creador orb enganyós* el viola a nivell de les realitzacions
 - El creador és enganyós perquè a simple vista, en detectar l'associació que modelitza, sembla que es compleix *Espill*; la violació és a nivell de les realitzacions, que no es mostren en els diagrames.

Principi, pàgina 168

Principi, pàgina 171

²⁶Si el model està ben construït (el que formalment en diem un model *coherent* →), tot enllaç present a MC es modelitza en forma d'enllaç dirigit, i tot enllaç dirigit és el model d'un enllaç. Per tant la visibilitat d'emmagatzematge no viola *Espill* ni a nivell d'abstracció ni a nivell de realitzacions.

14.3.1.Completesa i solidesa, pàgina 259

²⁷Considerem el model complet, → i per tant tot enllaç present a MC es modelitza en termes d'un enllaç dirigit.

14.3.1.Completesa i solidesa, pàgina 259

- Donat un *creador orb enganyós* la visibilitat d'emmagatzematge té dues funcions: modelitzar una associació i mantenir l'emmagatzematge. Però a nivell de les realitzacions aquesta dualitat no hi és present: hi ha enllaços dirigits que només modelitzen un enllaç; n'hi ha que mantenen un emmagatzematge; i n'hi ha que tenen les dues funcions

6.2.2 Creador canònic

Quan parlem de creador orb, a no ser que diguem el contrari, no estem pensant en els creadors canònics.

6.2.3 Creador canònic fals

Creador amb emmagatzematge selectiu. El creador basat en un emmagatzemador selectiu[→] només és capaç d'exercir la responsabilitat de la creació en un subconjunt del component objectiu. Això significa que:

- Els objectes realització del component objectiu O queden separats en dos grups: els que es poden crear a partir del component creador C escollit (els **generables**), i els que no (els **no generables**)
- Pel conjunt de realitzacions de O no generables a partir de C caldrà cercar un altre creador. Per tant, per un mateix component tindrem més d'un component creador simultàniament.
- El disseny es complica: el component creador depèn de l'objecte objectiu, i no pas del component del qual és realització.

Excursió. (Reconsideració de l'emmagatzemador selectiu) Suposem la visibilitat d'emmagatzematge $C \rightarrow O^*$. I suposem que C és un creador d'emmagatzemador selectiu de O . Sovint aquesta situació respon a una mala modelització. Si semànticament la qüestió té sentit, el que cal és considerar O com una generalització de A i B . Llavors C esdevindrà *creador general* de A (hem introduït l'obligatorietat), i ens manca trobar el creador de B . L'exemple 56[→] mostra com es pot fer aquest canvi en l'especificació.

Creador amb emmagatzemador general. El creador basat en un emmagatzemador general té, com hem vist,[→] dos problemes greus: el de la *guerra fratricida* (l'exercici d'una potencialitat per part d'algú elimina aquesta potencialitat de la resta), i el de la manca de simetria (no tots els objectes realització d'un mateix component tenen les mateixes propietats).

Res d'emmagatzemadors canònics falsos. Com a conclusió de les observacions anteriors direm que hem de fugir de *creadors canònics falsos*: compliquen el disseny (emmagatzemador selectiu) o introdueixen problemes semàntics que cal evitar (emmagatzemador general).

10.4.3.Principi de l'Emmagatzemador selectiu, pàgina 165

Exemple 56.Emmagatzemador general, pàgina 174

10.5.1.Creació i emmagatzematge, pàgina 173

Camins de fugida. Heus ací què cal fer davant un *emmagatzemador canònic fals*:

- *Emmagatzemador selectiu.* Podem intentar transformar MC per tal d'aconseguir un emmagatzemador general
- *Emmagatzemador general.* Si no podem trobar un creador canònic, és preferible quedar-nos amb un creador orb pur

6.2.4 Creador orb pur

Una proposta vàlida. Si no podem disposar d'un creador canònic aquesta és una opció vàlida: introduïm una visibilitat que viola *Espill* però amb una responsabilitat molt definida: l'emmagatzematge.²⁸ Per tant, els únics enllaços dirigits que tindrem seran enllaços dirigits d'emmagatzematge

Obtenció de l'emmagatzematge primari. Si a més prohibim les migracions externes i les comparticions internes, sabem que tot i violar *Espill*, l'emmagatzematge és un **emmagatzematge primari**.

6.2.5 Creador orb enganyós

Cohesió de l'emmagatzemador canònic. En el cas del *creador canònic* una mateixa visibilitat té dues responsabilitats: modelitzar una associació i mantenir l'emmagatzematge. Aquesta dualitat no provoca problemes perquè cada enllaç dirigit té ambdues responsabilitats. De fet, el que passa, és que es tracta d'una mateixa responsabilitat. Per tant, la cohesió és alta.

Un problema de baixa cohesió. En el cas de l'emmagatzemador orb enganyós, tenim també una visibilitat amb dues responsabilitats Però ara es tracta de dues responsabilitats diferents, no coincidents. Per tant la cohesió[→] és baixa: assignem dues responsabilitats que no tenen res a veure una amb l'altra a una mateixa visibilitat.

8.2.Cohesió, pàgina 127

6.2.6 Un parell de definicions

Unes definicions que seran útils. Tot seguit presentem unes definicions de *creador canònic* i *creador orb pur* que faciliten veure la diferència entre un i altre, i que ens seran força útils d'aquí endavant.

Definició. Creador canònic

Anomenem **creador canònic** l'origen de de la la visibilitat d'emmagatzematge, introduïda per la creació, que és el model de cap associació de MC, i tal que la seva hipotètica invers és **obligada i monoavaluada**.

²⁸ Així les coses, la visibilitat d'emmagatzematge compleix amb les condicions que el principi de la *Factorització pura* exigeix per poder violar *Espill*.

Definició. Creador orb pur

Anomenem *creador orb pur* l'origen de de la la visibilitat d'emmagatzematge, introduïda per la creació, que **no** és el model de cap associació de MC

7 Principi del Creador generalitzat

Principi



Creador generalitzat. Els candidats a *creador* són els *experts*, si cal per *per delegació*, de la creació que a més són *emmagatzemadors*. Per ordre, els emmagatzemadors que cal considerar són:

1. Emmagatzemador canònic (i llavors parlem de *creador canònic*)
2. Emmagatzemador orb pur (i llavors parlem de *creador orb (pur)*)

En el cas de *creació d'ús* (creació sense emmagatzematge) els candidats a *creador* són els *experts*, si cal *per delegació*, de la creació que a més en són *usuaris*.

Objectiu del creador generalitzat. El principi del *Creador generalitzat* expressa com s'han d'usar els principis de creació. De fet el que diu és que en aquest àmbit concret el principi de la *Simultaneïtat* → no és aplicable: els principis de creació s'han d'aplicar en ordre, i només quan no es pugui aplicar un principi intentarem aplicar el següent.

El guanyador. El *Creador canònic* → és el principi de creació per excel·lència. L'emmagatzematge proposat és un *emmagatzematge primari*, i no violem *Espill*, ni en les abstraccions ni en les realitzacions. Els creadors que proposa són *creadors canònics*. →

Un perdedor clar. Usar un *creador orb enganyós* queda fora de tota intenció. El model resultant té una cohesió molt baixa (com hem vist més amunt →); a part que la violació d'*Espill* per part d'alguns enllaços dirigits és difícil de justificar i complex de gestionar.

L'altre perdedor. El creador canònic fals també cal bandejar-lo: introdueixen dificultats semàntiques, compliquen el model.

Un recurs necessari. Les exigències del *creador canònic* són molt fortes (obligatorietat i monoavaluació de l'associació que modelitzem amb la visibilitat d'emmagatzematge), i no sempre es podran assolir. Per això cal admetre la possibilitat que, com a alternativa a la solució que no tenim, puguem usar una

Principi, pàgina 25

Principi, pàgina 175

Definició pàgina 181

10.6.2. Anàlisi del Creador orb, pàgina 178

visibilitat d'emmagatzematge no tant restrictiva. La proposta és cercar un **emmagatzemador orb pur** \rightarrow ja que si bé viola *Espill* (com el *creador orb enganyós*, que hem bandejat), manté una cohesió molt alta.

Els únics creadors orbs que ens interessin són els **creadors canònics** o els **creadors orbs purs**.

(27)

Notació.



- A partir d'ara, i mentre no diguem explícitament el contrari, un **creador orb** és un **creador orb pur**.
- Els **creadors canònics**, tot i ser també orbs, no els acostumarem a tractar a part

8 Migració i compartició de realitzacions

8.1 Presentació de les migracions

Concepte de migració. Sigui un enllaç dirigit $c : C \xrightarrow{atr} O^*$ tal que $o : O$ és una de les seves destinacions. Direm que $o : O$ **migra d'emmagatzematge** quan, pel motiu que sigui, $o : O$ deixa de formar part de les destinacions de $c : C \xrightarrow{atr} O^*$, per passar a formar part de les destinacions d'un altre enllaç dirigit $c' \xrightarrow{atr} O : O^*$.

Tipus de migracions. Distingim dos tipus d'emigracions:

- **Interna.** L'objecte que migra canvia d'objecte emmagatzemador, però no de visibilitat d'emmagatzematge

Abans: $o : O \in abast(c : C \xrightarrow{atr} O^*)$

Després: $o : O \in abast(c' : C' \xrightarrow{atr} O^*)$

- **Externa.** L'objecte que migra canvia d'objecte i de visibilitat d'emmagatzematge

Abans: $o : O \in abast(c : C \xrightarrow{atr} O^*)$

Després: $o : O \in abast(c' : C' \xrightarrow{atr} O^*)$

8.2 Presentació de les comparticions

Concepte de compartició. Direm que un objecte $o : O$ està **compartit** si pertany a dos emmagatzematges diferents. És a dir, si hi ha dos enllaços dirigits diferents tal que $o : O$ pertany a la destinació de tots dos.

Tipus de comparticions. Distingim dos tipus de comparticions:

- **Interna.** Els dos enllaços dirigits que tenen $\circ : \circ$ com una de les seves destinacions són realització d'una mateixa visibilitat

Sense: $o : O \in abast(c : C \xrightarrow{atr} O^*)$

Amb: $o : O \in abast(c : C \xrightarrow{atr} O^*)$
 $o : O \in abast(c' : C' \xrightarrow{atr} O^*)$

- **Externa.** Els dos enllaços dirigits que tenen $\circ : \circ$ com una de les seves destinacions són realització de visibilitats diferents

Abans: $o : O \in abast(c : C \xrightarrow{atr} O^*)$

Després: $o : O \in abast(c : C \xrightarrow{atr} O^*)$
 $o : O \in abast(c' : C' \xrightarrow{atr} O^*)$

8.3 Creador canònic, migracions i comparticions

El context de la migració. Suposem que com a resultat de la creació de \circ :
 \circ tenim $o : O \in abast(c : C \xrightarrow{atr} O^*)$. Si hem usat el principi *Creador canònic* sabem que $C \xrightarrow{atr} O^*$ és un emmagatzematge primari.

Migració externa. Si considerem l'emmagatzematge exigit pel *Creador canònic*, la **migració externa** no és possible.

Demostració.

- Suposem que $\circ : \circ$ ha patit una migració externa.
- La migració suposa un canvi d'enllaç dirigit d'emmagatzematge. Per tant $\circ : \circ$ que abans estava enllaçat a un objecte $: C$ (en concret $c : C$) ara deixa d'estar-ho. És a dir, la hipotètica inversa de la visibilitat $C \xrightarrow{atr} O^*$ no és obligada
- Com que la visibilitat $C \xrightarrow{atr} O^*$ s'ha introduït per *Creador canònic*, sabem que $C \xrightarrow{atr} O^*$ és el model d'una associació $C - O$ de MC. *Creador canònic* exigeix que aquesta associació sigui obligada per l'extrem de C . Per tant, donat $o : O$ tal que $model(o : O) = \circ : \circ$ sempre ha d'existir un enllaç $x : C - o : O$
- Abans de la migració aquest enllaç era $c : C - o : O$, on $model(c : C) = c : C$. Després de la migració hem de continuar tenint un enllaç $x : C - o : O$. Però aquest no està modelitzat en termes de cap enllaç dirigit. Per tant el model no és complert.
- Conclusió: La migració externa ens fa perdre les condicions exigides per l'emmagatzematge primari. Però com que l'emmagatzematge no viola *Espill* necessàriament ha de ser un emmagatzematge primari. Heus ací una contradicció.

□

Migració interna. Si considerem l'emmagatzematge exigít pel *Creador canònic*, la *migració interna* és possible.

Demostració.

- Suposem que $\circ : \circ$ ha patit una migració interna
- Això significa que $\circ : \circ$ continua pertanyent a l'abast d'un enllaç dirigit $x : C \xrightarrow{atr} \circ^*$ realització de $C \xrightarrow{atr} \circ^*$; el que ha canviat és quin és l'objecte $x : C$ origen d'aquest enllaç dirigit
- Abans de la migració $x : C$ és $c : C$; després és $c' : C$. En qualsevol dels dos casos $\circ : \circ$ només pertany a un enllaç dirigit realització de $C \xrightarrow{atr} \circ^*$
- Per tant, la migració interna manté les condicions exigides per l'emmagatzematge primari

□

Comparticions. L'emmagatzematge primari no permet **compartició interna**: la hipòtètica inversa de la visibilitat d'emmagatzematge no seria monoavaluada. Les *comparticions externes*, com que no afecten la visibilitat d'emmagatzematge, seran sempre possibles

Migracions, comparticions i creador canònic.

(28)

L'emmagatzematge exigít pel *creador canònic* és un *emmagatzematge primari*. Aquest emmagatzematge:

- Permet *migracions internes*
- No admet **migracions externes**
- No admet **comparticions internes**
- Permet *comparticions externes*

8.4 Creador orb pur, migracions i comparticions

L'emmagatzematge primari del Creador canònic. El *Creador canònic* introdueix un emmagatzematge que ha de complir les condicions d'emmagatzematge primari en tot moment. Això és conseqüència de la no violació d'*Espill* i de la completesa del model: l'associació que modelitza la visibilitat d'emmagatzematge és obligada i monoavaluada per l'extrem de C , i per tant també ho ha de ser el seu model en forma d'una hipòtètica visibilitat $\circ \rightarrow C$. Ho hem vist en detall una mica més amunt.[→] Tot seguit analitzem la situació pel cas del *Creador orb*.

Emmagatzemador orb pur. El *Creador orb* exigeix l'emmagatzematge, però permet que aquest violi *Espill*. Si considerem només els [emmagatzemadors orbs purs](#)[→] la visibilitat d'emmagatzematge que introdueix és, en principi, un [emmagatzematge primari](#):

- El component creador és creador de totes les realitzacions. (Tot-hom té un creador)
- Cada objecte només el creem un cop. (El creador és únic)

Migració externa. La visibilitat d'emmagatzematge introduïda pels [emmagatzemadors orbs purs](#) viola *Espill*. Per tant res impedeix que en algun moment un objecte $c : C$ pateixi una [migració externa](#). Simplement la visibilitat d'emmagatzematge emprada per la creació no serà un emmagatzematge primari. (Perdem l'obligatorietat de la hipotètica inversa).

Migració interna. La [migració interna](#) tampoc porta problemes. En aquest cas, però, es mantenen les condicions de l'emmagatzematge primari. La situació és la mateixa que hem analitzat pel cas del *Creador canònic*.[→]

Compartició. A l'igual que pel cas del *Creador canònic*,[→] la [compartició externa](#) no porta problemes, però la compartició interna impedeix que la visibilitat d'emmagatzematge sigui un emmagatzematge primari

Comparticions. L'[emmagatzematge primari](#) no permet **compartició interna**: la hipotètica inversa de la visibilitat d'emmagatzematge no seria monoavaluada. Les [comparticions externes](#), com que no afecten la visibilitat d'emmagatzematge, seran sempre possibles

(29)

Migracions, comparticions i *creador orb pur*.

L'emmagatzematge exigit pel [creador orb pur](#) és un [emmagatzematge primari](#). Aquest emmagatzematge:

- Permet [migracions internes](#)
- Permet [migracions externes](#)
- Permet [comparticions internes](#)
- Permet [comparticions externes](#)

²⁸Estem considerant un emmagatzematge que viola *Espill*. Per tant els enllaços dirigits que introduïm per l'emmagatzematge són els únics que existeixen. Com que la visibilitat d'emmagatzematge no és el model de cap associació no hi ha enllaços presents a MC que per completesa del model s'hagin de modelitzar en termes d'enllaços dirigits.

Emmagatzematge primari i creador orb.

(30)

Si a l'emmagatzematge exigít pel *creador orb* li prohibim la migració externa és i les comparticions internes, obtenim un *emmagatzematge primari*

8.5 Creació i emmagatzematge primari

Emmagatzematge primari en la creació.

(31)

Davant l'absència de migracions externes i de comparticions internes, l'emmagatzematge necessari per a la creació és un *emmagatzematge primari*

Com assegurem l'emmagatzematge primari. A partir de les conclusions dels apartats anteriors tenim que:

- Si usem principi del *Creador canònic* es compleixen totes les condicions necessàries per assegurar que tenim emmagatzematge primari
- Si usem principi del *Creador orb*²⁹ per assegurar que tenim emmagatzematge primari cal imposar les condicions sobre migracions i comparticions

9 Ús dels diferents principis de creació

9.1 Jerarquització dels principis de creació

Resistència a trencar la Modelització continua. L'ús com a creador d'un emmagatzemador no canònic introdueix una visibilitat d'atribut a MComp que no es correspon a cap associació de MC.³⁰ Això viola el principi de la Modelització continua, principi que considerem bastant fort.³¹ Per això davant la possibilitat d'un emmagatzemador canònic, aquest el considerarem com a candidat prioritari.

²⁹Recordem que el principi *Creador generalitzat* \rightarrow ens diu que l'únic creador orb vàlid és el *creador orb pur*.

³⁰Un emmagatzemador no canònic també pot ser que introdueixi enllaços dirigits que violin *Espill*. Aquí només parlem de visibilitats per simplificació expositiva.

³¹De fet el principi de la Modelització continua gairebé el volem usar com una regla d'obligat compliment, violable només sota el paraigua del principi de la *Fabricació pura* o sota d'altres raons fortes del disseny.

Delegació com a segona opció. La *delegació* permet estendre els *creadors* considerats, a base de delegar l'expertesa, i amb el preu d'acoblar els intermediaris a la informació de l'expertesa. Per tant considerarem com a prioritari tot candidat expert directe per sobre dels experts per delegació.

Els *usuaris* com a recurs d'emergència. L'ús d'*usuaris* com a *creadors* dóna inestabilitat al disseny, i per tant també intentarem evitar-ho sempre que sigui possible. De fet només els contemplarem quan no tinguem cap altra possibilitat.

Preponderància de *Creador canònic*. De tot plegat en podem concloure que el principi *Creador canònic*, que demana *emmagatzemadors* (fuig del recurs dels *usuaris*), que exigeix que siguin *canònics* (manté la *Modelització continua*), i que permet la *delegació de l'expertesa* (tot i que només en segona opció) es presenta com el principi bàsic per aplicar en un cas de creació. I això és justament el que postula el principi del *Creador generalitzat*. →

Principi, pàgina 182

Violació del principi d'Igualtat d'oportunitats. Els principis de la *Simultaneïtat* i de la *Igualtat d'oportunitats* diuen que cal aplicar simultàniament els principis *Expert*, *Creador (GRASP)*, *Creador delegat*, *Creador orb* i *Creador canònic*; i que cal considerar tots els creadors obtinguts. En aquest cas, però, i tal com acabem de comentar, les propostes del principi del *Creador canònic* es consideren de més rellevància que les propostes dels altres principis.

9.2 Candidats no considerats amb l'ús del *Creador canònic*

Anàlisi de pèrdues. Tot seguit analitzem quins són els candidats a creador que perdem si només apliquem el principi *Creador canònic*.

Candidats no contemplats pel principi *Creador canònic*. Els candidats que poden aportar els altres principis, i que no aporta *Creador canònic*, no són tants com sembla: *usuaris*, *emmagatzemadors no canònics*, i aquells *experts* que no són ni *usuaris* ni *emmagatzemadors*.

Què perdem amb els *usuaris*. En l'anàlisi dels *usuaris* feta més amunt → hem justificat que donar el rol de *creador* als *emmagatzemadors* és preferible a donar-lo als *usuaris*; i el cas en què no hi ha *emmagatzematge (creació d'ús)* ja el contempla el principi *Creador orb*. Per tant no estem perdent cap candidat interessant.

Què perdem amb els *experts* que no són ni *usuaris* ni tampoc *emmagatzemadors*. Si donem el rol de *creador* a un component amb aquestes característiques, tenim que l'únic objectiu de la creació és la comunicació. I això ho prohibeix *Creador (GRASP)* i *Baix Acoblament*. Per tant en aquest cas tampoc perdem cap candidat interessant.

Què perdem amb els emmagatzemadors no canònics. Aquest és l'únic cas en què *Creador orb* (i per tant també *Creador delegat*, i potser també *Creador (GRASP)*) poden oferir un candidat no contemplat per *Creador canònic* i que podria ser interessant considerar. L'acceptació d'un candidat així, però, exigeix modificar MC, o bé justificar la violació de la *Modelització continua*. Sigui com sigui es tracta d'un supòsit força extrem i especial, i per tant té sentit que, en primera instància, no el considerem.

Quan cal recórrer als emmagatzemadors no canònics. Són tres les circumstàncies en les que cal recórrer a *emmagatzemadors no canònics*:

1. No tenim cap candidat a *creador canònic*
2. Els dissenys resultant de considerar els diferents *creadors canònics* no ens satisfan (en termes, per exemple, d'*acoblament* i *cohesió*)
3. Ens interessa analitzar dissenys alternatius, sigui per donar força a la nostra proposta, sigui per qüestions d'eficiència³² o per qualsevol altre motiu.

Ús dels principis de creació.

Davant d'una necessitat de creació, el principi del *Creador generalitzat* → diu que cal aplicar el principi del *Creador canònic*; i segons les circumstàncies també el principi del *Creador orb (pur)*.³³

(32)

Principi, pàgina 182

10 La patata calenta i els principis de creació

Expertesa en la creació. L'*Expert* ens pot donar més d'un creador; l'*Expert* es fixa només en la informació necessària per fer la creació.

Comunicar no pot ser un rol bàsic. L'anàlisi de què fem amb la nova realització (el problema de la *patata calenta*) ens diu que en termes d'*acoblament* la possibilitat de crear per comunicar no és una bona opció.

Coincidència de rols. El principi *Creador (GRASP)* → captura les idees anteriors en exigir que el rol de *creador* coincideixi amb el d'*emmagatzemador* i/o *usuari*.

Principi, pàgina 158

Delegació de l'expertesa. Per tal de poder aconseguir la coincidència ens cal relaxar les condicions de *Creador (GRASP)* en el sentit que el *creador* no cal que sigui un *expert* en la creació, sinó que n'hi ha prou en què en sigui un *expert delegat*. El resultat és el principi *Creador delegat*. →

Principi, pàgina 160

³²El principi de l'*Eficiència en segon pla* → ens diu que els aspectes d'eficiència només han de ser considerats en un segon pla, mai en primera instància.

12.3.2. Perill de l'eficiència prematura, pàgina 228

³³Les circumstàncies d'aplicació del principi del *Creador orb* són les expressades en el paràgraf anterior.

Independència dels usuaris. La dificultat de detectar els *usuaris futurs*, la *inestabilitat del disseny* (contemplada en els principis de la *Teranyina* i de la *Reconsideració*), i la pretesa *estabilitat de l'especificació*, porta a plantejar-se com a candidats a *creadors* només els *emmagatzemadors* (en el cas que n'hi hagi). Les conclusions les expressa el principi *Creador orb*. →

Principi, pàgina 163

La independència és Cohesió. El principi *Creador orb* es pot veure com una restricció del principi *Creador delegat* que aconsegueix una major *Cohesió*.

Emmagatzematge i MC. La *Modelització continua* exigeix que els enllaços induïts per l'emmagatzematge coincideixin amb enllaços exigits pel MC. El principi *Emmagatzemador (canònic)* → dóna els candidats a *emmagatzemador* que compleixen aquesta condició.

Principi, pàgina 172

Orientació de les visibilitats. L'emmagatzematge d'una nova realització significa una visibilitat *incident* sobre l'objectiu.

Creació i emmagatzematge canònic. La restricció als emmagatzemadors que no violen la *Modelització continua* converteix el principi *Creador orb* en el principi *Creador canònic*. →

Principi, pàgina 175

Principi preponderant. L'estabilitat del disseny i la no violació del principi de la *Modelització continua* ens porten a considerar les propostes del principi *Creador canònic* com a més rellevants que les propostes dels altres principis, fet que constitueix una violació del principi de la *Igualtat d'oportunitats*. Aquesta preponderància queda explicatada en el principi del *Creador generalitzat*. →

Principi, pàgina 182

Emmagatzemadors no admesos. En el cas de no poder (o no voler) disposar d'un emmagatzemador canònic, no tot emmagatzemador és vàlid. En concret només podem considerar els *emmagatzemadors orbes purs*, i això és el que diu el principi del *Creador generalitzat*. →

Principi, pàgina 182

Conseqüències de la violació de la Igualtat d'oportunitats. Malgrat les estrictes condicions que imposa, l'ús exclusiu de *Creador canònic* no deixa fora de joc candidats interessants. Els únics candidats a creador que potser caldria considerar a més a més són els emmagatzemadors no canònics, però com que la seva acceptació viola el principi de la *Modelització continua* ens veuríem obligats a justificar aquesta violació o bé hauríem de modificar MC.

Anàlisi a dos nivells. En conclusió, l'únic principi que cal aplicar és el *Creador canònic*, tot i que hem de deixar la porta oberta a afegir com a possible candidat a *creador* els proposats per *Creador orb (pur)*: expert, directe o per delegació, que sigui un emmagatzemador, i tal que la visibilitat d'emmagatzematge resultant no sigui el model de cap associació de MC.

Simplificació de nomenclatura. Tot plegat ens permet simplificar la nomenclatura de la manera que segueix:

Notació. A partir d'ara quan parlem del principi *Creador* ens referirem al principi *Creador canònic*. Allà on sigui necessari diferenciar, usarem els termes principi *Creador (GRASP)*, principi *Creador delegat*, principi *Creador orb* i principi *Creador canònic*.



Notació. A partir d'ara quan parlem del principi *Creador orb* ens referirem al principi *Creador orb pur*.



11 Creació i recuperació de la informació

11.1 Anàlisi de la recuperació en una creació

Informació d'expertesa en la creació. La *informació d'expertesa en la creació* d'un objecte \circ és part *essencial* d'aquest objecte: sense la informació d'expertesa no s'hauria pogut crear \circ ; sense la informació d'expertesa l'objecte \circ no podria existir.

Propietats de l'objecte creat. La informació d'expertesa usada en la creació d'un objecte forma part de les *propietats* d'aquest objecte. És en aquest sentit que la informació d'expertesa en la creació forma part de l'*essència* d'aquest objecte.

Recuperació de l'expertesa de la creació. El disseny s'haurà de preocupar, per cada creació resolta, d'assegurar que es pot recuperar la informació d'expertesa emprada en la seva creació.

Maneres de recuperar la informació d'expertesa. Donat un objecte $\circ : \circ$ la informació d'expertesa emprada en la seva creació es pot obtenir de dues maneres diferents:

1. *Explícitament.* La informació d'expertesa per a la creació apareix a la interfície del component

Per exemple, el component C pot oferir una operació *recuperaExpertesa()* : *<Info expertesa>* que permet, donat un objecte qualsevol que n'és realització, recuperar-ne la informació d'expertesa emprada en la seva creació.

2. *Implícitament.* Per accedir a un objecte cal saber prèviament quina informació es necessitaria per a la seva creació.

D'alguna manera el disseny assegura que si tenim accés a un objecte $\circ : \circ$ necessàriament, i prèviament, tenim accés a la informació d'expertesa emprada en la seva creació.

Responsabilitats del disseny. És responsabilitat del disseny assegurar que per a tot objecte $o:O$ la informació d'expertesa emprada en la seva creació és recuperable, sigui *explícitament* o sigui *implícitament*.

Exemple 57 (Recuperació en la creació) La informació d'expertesa per a la creació d'una $i:Inscripció$ és la $c:Caminada$ a la que es refereix, i la data en la que es realitza la $i:Inscripció$.

Donada una $i:Inscripció$ pot ser útil conèixer en quina data es va realitzar. Això exigeix que el disseny ha d'assegurar-ne la seva recuperació per a tota $i:Inscripció$.

Qui és però el responsable de recuperar la data d'una $i:Inscripció$? Per Expert, és la pròpia *Inscripció* qui rebrà aquesta responsabilitat. Per tant afegim al component *Inscripció* l'operació `recuperaData() : <Data>` que retorna la data en la que es realitzà la inscripció.

I qui és el responsable de recuperar la $c:Caminada$ a què es refereix una $i:Inscripció$? El mateix raonament fet en el cas de la data ens porta a afegir al component *Inscripció* l'operació `recuperaCaminada() : <Caminada>` que retorna la $c:Caminada$ pertinent.

Per tant, tant per a la data com per a la $c:Caminada$ tenim un mecanisme de recuperació explícit.

Si bé en el cas de la data sembla clar que la recuperació explícita és l'únic mecanisme possible, el cas de la $c:Caminada$ necessita una anàlisi més acurada. Com veurem al capítol 11.^{GLS} L'ES novaInscripció, → segons qui sigui el creador d' *Inscripció* pot ser que l'única manera d'accedir a una $i:Inscripció$ determinada sigui a partir de la $c:Caminada$ corresponent. En aquest cas no cal que la *Inscripció* tingui un mètode que faci pública la $c:Caminada$ a la que es refereix. I per tant el mecanisme de recuperació implícita és més que suficient.³⁴

11.GLS L'ES novaInscripció, pàgina 199

11.2 Excursió: Recuperació d'informació

L'anàlisi de la recuperació d'informació a partir de la informació necessària per a la creació és un cas particular d'una necessitat més general: la recuperació de les propietats d'un component. Tot seguit fem un esbós de les idees més rellevants.

11.2.1 La informació en el MC

Atributs dels conceptes. Els conceptes³⁵ poden tenir *atributs*: és informació que exigim que tingui cada realització dels conceptes.

??, pàgina ??

³⁴Al capítol ??, → introduïrem els principis de disseny que ens ajuden a determinar, en cada cas, si la recuperació de l'expertesa en la creació ha de ser explícita o implícita.

³⁵Compte: hem retrocedit a l'especificació; estem en el MC.

Exemple 58 (Concepte amb atributs)

- *Identificadors.* Els identificadors dels conceptes són atributs d'aquests. Sovint, però, i quan el context és prou clar, no s'acostumen a plasmar en el diagrama del MC.
- *Atributs obligats.* En l'exemple del TPV un atribut de la línia de venda és la quantitat d'unitats venudes en aquesta línia. Es tracta d'una informació obligatòria o essencial per a la línia: no es pot crear una línia sense indicar el nombre d'unitats del producte venut.
- *Atributs potestadius.* Un atribut de Participant és la seva edat. Aquest atribut, però, no és obligatori: de fet l'únic que ens interessa és assegurar que el responsable és major d'edat; no ens cal saber la seva edat. Malgrat tot, el coneixement de l'edat dels participants permet fer estadístiques de participació, i permet afinar en els obsequis donats. Per tant és concebible que el sistema mantingui participants dels quals en desconexem l'edat.

Atributs i associacions. Les associacions que un concepte té amb d'altres conceptes també són propietats d'aquest concepte, a l'igual que els atributs.

Exemple 59 (Atributs i associacions)

- *Associacions obligades.* Una propietat essencial i obligatòria de les inscripcions és la caminada a la que es refereixen. Una propietat essencial i obligatòria de les línies de venda és el producte que es ven en aquesta línia. En ambdós casos es tracta de propietats expressades en termes d'associacions obligades.
- *Associacions optatives.* Una de les propietats dels participants és l'obsequi rebut en acabar la caminada a la que s'han inscrit. Però no tots els participants tenen obsequi: només els qui han acabat la caminada. Aquesta propietat es modelitza com una associació optativa.

Excursió. (El paper dels atributs) El paper de les associacions i els atributs depèn del mètode d'especificació emprat.

- **Sobreespecificació.** Per alguns mètodes d'especificació només hi ha atributs. L'ús d'aquests mètodes exigeix que l'especificació s'ha de fer responsable de la direccionalitat de les associacions, la qual cosa, des nostre punt de vista, constitueix una sobreespecificació.³⁶

Per nosaltres la direccionalitat és una necessitat pel mecanisme dels missatges, i per tant ha de ser responsabilitat del disseny. L'especificació ha de plasmar la interrelació entre els conceptes, però no ha d'entrar en com interactuen i col·laboren entre ells.

³⁶Exemple. El que nosaltres modelitzem com una associació $f:A \rightarrow B$ caldria modelitzar-la o bé amb un atribut $f:A$ de B (hem orientat f de B a A), a o bé amb un atribut $f:B$ de A (hem orientat f de A a B).

- **Sobrediaqramació.** Per alguns mètodes d'especificació només hi ha *associacions*. L'expressivitat d'aquests mètodes és equivalent a la que proposem aquí, però el seu ús omple excessivament els diagrames de conceptes; al nostre entendre es tracta d'una *sobrediaqramació*.

L'avantatge d'aquests mètodes és la simplicitat, en no haver de distingir entre *atribut* i *associació*.

- **Dos mecanismes.** La nostra proposta no assigna responsabilitats indegudes a l'especificació, i no carrega excessivament els diagrames. La contrapartida és la necessitat de distingir entre *atribut* i *associació*.

En principi considerarem com a *atribut* tota propietat expressable amb un *tipus bàsic*, tot i que la recomanació és de no expressar en l'especificació quin és exactament aquest tipus bàsic.³⁷

11.2.2 La informació a MComp

Atributs que modelitzen atributs. Si *A* és un concepte que té un atribut *atr*, i *A* és el component que modelitza el concepte *A*, llavors el component *A* conté un *atribut del disseny* de nom *atr* que modelitza l'atribut *atr* de *A*.

Atributs del disseny. En el disseny poden aparèixer *atributs* que no modelitzen cap propietat dels conceptes associats; i que per tant violen el principi de l'*Espill*. És el cas, per exemple, de resultats parcials o temporals d'algun càlcul.

Visibilitat d'atribut. Les *visibilitats d'atribut* són el mecanisme del disseny per modelitzar associacions de MC, per emmagatzemar objectes, i per expressar la potencialitat de col·laboració. Una *visibilitat d'atribut* no és un *atribut del disseny*.

Excursió. (Atributs i visibilitat d'atribut) La visió dels atributs com un cas més d'associació (com fan els mètodes d'especificació que només consideren les associacions) és exportable al disseny: els components tenen visibilitats d'atribut, però no pas atributs.

En les propostes, com la que seguim nosaltres, en les que el disseny distingeix entre les *visibilitats d'atribut* i els *atributs de disseny*,³⁸ és la implementació qui pot decidir-se per la convergència: la implementació de les visibilitats d'atribut d'un component com un *atribut d'implementació* de la *classe* que modelitza el component elimina, a nivell d'implementació, tota diferència³⁹ entre un atribut i una visibilitat d'atribut.

³⁷El problema però no s'ha evitat: queda per definir què és un tipus bàsic. Ho és una *paraula* (*string* o similar)? Ho és una *data*?. La proposta és força ambigua en aquest aspecte (i en certa manera circular): considerem tipus bàsic tot aquell que no ens cal com a concepte.

³⁸Aquesta distinció és coherent amb la distinció feta a l'especificació.

³⁹En llenguatges OOP certament no hi ha cap diferència; en llenguatges híbrids hi poden haver diferències d'emmagatzematge intern; per exemple, en C++ un atribut potser és emmagatzemat com a valor, mentre que una visibilitat d'atribut requerirà un atribut emmagatzemat com a apuntdor.

Propietats atributives. En alguns aspectes del disseny no és tant rellevant si una determinada propietat es manté en termes d'atribut o en termes de visibilitat d'atribut, com el fet que es manté la propietat. En aquests casos enlloc de parlar d'atribut o de visibilitat d'atribut parlarem de *propietat atributiva*. →

11.2.3 Utilitat de la informació en el disseny

Utilitat de la informació. És responsabilitat del disseny assegurar que les *propietats atributives* dels elements usats per a la modelització són *significatives* en algun sentit. En concret, tota propietat atributiva d'un element o bé ha de ser *consultable*, o bé és *usada* per algú per tal de decidir-se per un comportament o per un altre.⁴⁰

Utilitat en el disseny. L'exigència d'utilitat dels elements usats per construir els models del disseny significa que, com a mínim:

- Tota *visibilitat d'atribut* s'ha d'usar en alguna col·laboració
- Tot component ha d'oferir mecanismes per a la consulta dels seus *atributs*

11.2.4 Anàlisi de la recuperació

Procés d'anàlisi de la recuperació. El *procés d'anàlisi de la recuperació* ha de decidir quines de les propietats atributives de cada component són visibles a la interfície d'aquest; i amb quin mecanisme.⁴¹

Recuperació dels atributs. El *procés d'anàlisi de la recuperació* haurà de decidir quins dels *atributs* de cada component es fan públics a través de la interfície.

Recuperació de les destinacions de les visibilitats d'atribut. El *procés d'anàlisi de la recuperació* també haurà de decidir si els components ofereixen en la seva interfície mecanismes per a la recuperació de les destinacions dels enllaços dirigits corresponents a les visibilitats d'atribut.⁴²

Paràmetres i visibilitat de paràmetre. La publicació de propietats atributives a la interfície d'un component significa que aquest ha de tenir operacions que retornin aquesta propietat. En el cas que el que es publica sigui l'objectiu d'una *visibilitat d'atribut* el mecanisme de recuperació significa afegir una *visibilitat de paràmetre* coherent amb aquesta *visibilitat d'atribut*. Si el que es recupera és un *atribut*, podem parlar de *paràmetre o valor de retorn*, però pròpiament no estem davant d'una *visibilitat de paràmetre*.⁴³

⁴⁰La mateixa afirmació s'ha fet, tot i que en altres termes i amb una altra intenció, a l'apartat 4.4. *Consistència dels models*, pàgina 84.

⁴¹Entenem per mecanisme el mètode o conjunt de mètodes que permet que una propietat aflori a la interfície del component.

⁴²D'aquí l'expressió "com a mínim" usada en el paràgraf *Utilitat en el disseny*: no només els atributs han de ser recuperables, sinó també les destinacions de les visibilitats d'atribut.

⁴³La següent excursió aprofunditza i justifica aquesta darrera afirmació.

Excursió. (Paràmetres i visibilitat de paràmetre) En el cas de només admetre visibilitats d'atribut, la recuperació sempre significa introduir una visibilitat de paràmetre coherent amb aquesta visibilitat d'atribut.

Si seguim la proposta de distingir entre atributs i visibilitat d'atributs, cal llavors també distingir entre paràmetre i visibilitat de paràmetre (tot i que potser la distinció desapareixerà a nivell d'implementació).

Una alternativa per evitar aquesta complicació addicional seria:

- En l'especificació distingim entre atributs i associacions, però només des d'un punt de vista sintàctic: per nosaltres són la mateixa cosa, però per simplicitat dels diagrames usem els atributs allà on ens convingui
- En el disseny mantenim la visió unitària (només tenim visibilitats) tot i que per simplicitat en els diagrames deixem expressar determinades visibilitats en forma d'atributs

Principis i definicions del capítol

Principis

Creador (*GRASP*), 158
Creador canònic, 175, 178
Creador delegat, 160
Creador generalitzat, 182
Creador orb, 163

Emmagatzemador (canònic),
172
Emmagatzemador general,
171
Emmagatzemador selectiu,
168

Definicions

Creador canònic, 176, 181
Creador orb, 163
Creador orb pur, 182

Emmagatzemador canònic,
172
Emmagatzemador general,
171
Emmagatzemador selectiu,
169
Emmagatzematge primari,
176
Expert per delegació, 159

Informació d'expertesa, 159

Capítol 11

GLS L'ES novaInscripció

1	La i : Inscripció com a patata calenta	202
1.1	Experts en la creació	202
1.2	Creació amb emmagatzematge	202
1.3	Creador canònic	202
1.4	Creador orb	203
1.5	M1: El controlador (gls:GLS) com a creador	203
1.6	M2: La c:Caminada com a creador	208
1.7	Comparativa dels dos possibles models	213
1.8	Validesa de les decisions	214
2	POST: Enllaçar i i c	215
2.1	M1: el controlador com a a creador de i	215
2.2	M2: la c:Caminada com a creadora de i	215

Contingut detallat del capítol 11

1 La i:Inscripció com a patata calenta	202
1.1 Experts en la creació	202
1.2 Creació amb emmagatzematge	202
Persistència entre ES	202
Excursió. (Realització activa)	202
Persistència entre CU	202
Creador emmagatzemador	202
Multiplicitat de la visibilitat d'atribut	202
1.3 Creador canònic	202
Condicions per <i>Creador canònic</i>	202
Associacions de MC	202
Condicions per a l'emmagatzematge canònic	203
Emmagatzemador canònic	203
Creador canònic	203
1.4 Creador orb	203
Altres possibles creadors	203
Creadors orbs	203
Creador orb pur	203
Exercici pedagògic	203
1.5 M1: El controlador (gls:GLS) com a creador	203
1.5.1 Aspecte general del model	203
Encarrilament	203
Enllaços dirigits d'emmagatzematge	204
Associació hipotètica	204
1.5.2 Visibilitat d'emmagatzematge	204
Necessitats d'emmagatzematge	204
Destinació de l'enllaç dirigit d'emmagatzematge	204
Multiplicitat de la visibilitat d'emmagatzematge	204
Totes les Inscripcions juntes	204
Repositori centralitzat	204
1.5.3 El multiobjecte com a objecte	205
Modisme MOMO sobre els multiobjectes: recordatori	205
Aplicació de MOMO	205
Creador del multiobjecte d'Inscripcions	205
Excursió. (Creació primerenca)	205
1.5.4 Recuperació de la informació	207
Recuperació explícita de la Caminada	207
Accés amb desconeixement	207
Responsabilitat del manteniment de la Caminada	207
Recuperació explícita de la Caminada	207
Manteniment de la Caminada	207
Multiplicitat de la visibilitat de recuperació	207
Creació de l'enllaç dirigit i:Inscripció ^{atr} c:Caminada	207

	Excursió. (Opcionalitat de l'emmagatzematge)	208
	No cal una POST explícita	208
1.5.5	Anàlisi de l'acoblament de $M1$	208
1.6	$M2$: La c:Caminada com a creador	208
1.6.1	Aspecte general del model	208
	Enllaços dirigits d'emmagatzematge	208
	Associació <i>referent a</i>	208
1.6.2	Visibilitat d'emmagatzematge	209
	Necessitats d'emmagatzematge	209
	Destinació de l'enllaç dirigit d'emmagatzematge	209
	Multiplicitat de la visibilitat d'emmagatzematge	209
	Inscripcions agrupades per Caminada	209
	Repositori distribuït per Caminades	209
1.6.3	El multiobjecte com a objecte	210
	Modisme <i>MOMO</i> sobre els multiobjectes: recordatori	210
	Aplicació de <i>MOMO</i>	210
	Creadors dels multiobjectes d'Inscripcions	210
	Excursió. (Creació primerenca)	210
1.6.4	Recuperació de la informació	211
	Possibilitat de recuperació implícita de la Camianda	211
	Accés amb coneixement	211
	Accés a les Inscripcions	211
	Inscripcions compartides	211
	Ús d'un repositori alternatiu	211
	Simplicitat	211
	Excursió. (Opcionalitat de l'emmagatzematge)	212
1.6.5	Anàlisi de l'acoblament de $M2$	212
1.7	Comparativa dels dos possibles models	213
	Anàlisi de l'acoblament	213
	Anàlisi de la cohesió	213
	Visibilitat d'emmagatzematge	213
	Modelització de l'associació "es refereix"	213
1.8	Validesa de les decisions	214
	Interpretació de les hipòtesis	214
	Escenaris alternatius	214
	Validesa condicionada del disseny	214
2	POST: Enllaçar i i c	215
2.1	$M1$: el controlador com a creador de i	215
2.2	$M2$: la c:Caminada com a creadora de i	215

1 La i:Inscripció com a patata calenta

1.1 Experts en la creació

Tenim dues alternatives per assignar la responsabilitat de crear la nova i: tant gls:GLS com c:Caminada en són *experts*.[→]

7.3.3.Expert, pàgina 122

1.2 Creació amb emmagatzematge

Persistència entre ES. La nova i:Inscripció té una vida superior a la de l'ES que ha demanat la seva creació: en concret, els altres ES del mateix CU volen actuar sobre aquesta i:Inscripció. Per tant, cal emmagatzemar-la.

Excursió. (Realització activa) Una de les POST encara no analitzades de l'ES *novaInscripcio* (c:Caminada) demana que la nova realització i:Inscripció es mantingui activa. La semàntica d'aquesta exigència[→] és justament la de comunicar la i:Inscripció entre els diferents ES del mateix CU. El mecanisme per mantenir aquesta semàntica és l'emmagatzematge: cal una cadena de visibilitats d'atribut des del controlador fins a la i:Inscripció activa.

12.Arguments implícits, pàgina 217

Persistència entre CU. La nova i:Inscripció fins i tot té una vida superior a la del CU que ha demanat la seva creació. Per exemple, el CU *ferDorsals* ha d'actuar sobre les inscripcions creades anteriorment; i justament aquesta creació és la responsabilitat del CU *ferInscripcio*. Per tant, cal emmagatzemar la i:Inscripció.

Creador emmagatzemador. Tant el principi *Creador canònic* com *Creador orb* exigeixen que davant una creació amb emmagatzematge com la que ens trobem, el creador sigui també emmagatzemador. I això significa una visibilitat d'atribut des del creador a l'objectiu: $\text{creador} \xrightarrow{\text{atr}} \text{objectiu}^*$.

Multiplicitat de la visibilitat d'atribut. La visibilitat d'atribut que imposa l'emmagatzematge és una visibilitat generalitzada. Per decidir si es tracta d'una visibilitat mono o multiavaluada caldrà prendre en consideració d'altres aspectes: la multiplicitat de l'associació que modelitza, les necessitats d'emmagatzematge múltiple i les necessitats de col·laboració amb diferents objectes.

1.3 Creador canònic

Condicions per Creador canònic. Per tal d'aplicar el principi *Creador canònic* cal cercar en el MC les associacions d'*Inscripció* amb els possibles *experts* (si cal per delegació) de la creació. Comencem pels *experts directes*; i si no ens en sortim ja analitzarem les possibles delegacions de l'expertesa.¹

¹En aquest cas, però, tots els components que tenim de moment a MComp són experts directes en la creació.

Associacions de MC. Per tant, cal analitzar si en el MC hi ha alguna associació entre *GLS* i *Inscripció*; o entre *Caminada* i *Inscripció*. Veiem que efectivament entre *Caminada* i *Inscripció* hi ha l'associació “referent *a*” ($1 - N$), mentre que entre *GLS* i *Inscripció* no hi ha cap associació.

Condicions per a l'emmagatzematge canònic. Per tal que *Caminada* pugui ser un *emmagatzemador canònic* cal assegurar que la visibilitat d'emmagatzematge $\text{Caminada} \xrightarrow{\text{atr}} \text{Inscripció}^*$ és el model d'una associació *Caminada* – *Inscripció*, obligada i monoavaluada a l'extrem de *Caminada*.

Emmagatzemador canònic. L'associació “referent *a*” compleix les condicions exigides: tota *inscripció* es refereix a una *caminada* (obligatorietat) i només a una (monoavaluació). Per tant, podem considerar la *Caminada* com *emmagatzemador canònic* de la *i*:*Inscripció* acabada de crear.

Creador canònic. *Caminada* és *experta* en la creació de la *i*:*Inscripció*; i també n'és *emmagatzemador canònic*. Per tant, pel principi de *Creador canònic* *Caminada* és la *creadora* d'*Inscripció*.

1.4 Creador orb

Altres possibles creadors. Malgrat la preeminència del principi *Creador canònic*, cal deixar la porta oberta a la possibilitat d'un creador que no sigui *emmagatzemador canònic*. Cal recordar però que el principi del *Creador generalitzat* ens diu que ens hem de limitar a *creadors orbs purs*.

Creadors orbs. En el cas que ens ocupa, l'única possibilitat diferent de *Caminada* és el controlador *GLS*. Això és així perquè a MC no tenim cap altre concepte, a part de *Caminada* i *Inscripció*; i a MComp només tenim els components *Caminada*, *Inscripció* i *GLS*.

Creador orb pur. El controlador *GLS* compleix les condicions d'*emmagatzemador orb pur*, ja que en el MC no hi ha cap associació *GLS* – *Inscripció*. I com que és *expert* en la creació, és un *creador orb pur*.

Exercici pedagògic. Per tal de poder copsar les conseqüències d'usar un creador canònic o no, en el que segueix desenvoluparem els dos models (*M1* segons el *creador* proposat per *Creador orb*,² i *M2* segons el *creador* proposat per *Creador canònic*, tot i que a priori tenim clar que ens hem de decantar pel model que considera la *Caminada* com a *creadora* (*M2*).

1.5 *M1*: El controlador (*gls*:*GLS*) com a creador

1.5.1 Aspecte general del model

Encarrilament. Analitzem el cas en què prenem *gls*:*GLS* com a creador de la *i*:*Inscripció*. Tenim que *gls*:*GLS* l'hem escollit com a controlador; per

²Recordem aquí que, si no es diu el contrari, quan parlem de *creador orb* ens referim a un *creador orb pur*.

tant el controlador té dues tasques assignades: la captura dels esdeveniments de sistema i la creació d'una nova *Inscripció*. Aquestes dues responsabilitats no semblen massa relacionades, i per tant podem parlar d'una *Baixa cohesió* del controlador. Precisament el principi de l'*Encarrilament* resol la situació demanant que el controlador delegui totes les possibles responsabilitats.

Enllaços dirigits d'emmagatzematge. El *controlador* és creador i emmagatzemador de la *i:Inscripció*. L'emmagatzematge exigeix el manteniment d'un enllaç dirigit (generalitzat) d'atribut entre el controlador i la nova *i:Inscripció* creada: $gls:GLS \xrightarrow{atr} Inscripció^*$, tal que *i:Inscripció* formi part de la destinació d'aquest enllaç dirigit. \rightarrow

9.3.3.Emmagatzematge, pàgina 141

Associació hipotètica. Així, l'ús del controlador com a *creador* ens porta a mantenir uns enllaços dirigits que no es corresponen a cap associació de MC. En aquest cas caldria modificar MC, o justificar el perquè de la violació del principi de l'*Espill*.

1.5.2 Visibilitat d'emmagatzematge

Necessitats d'emmagatzematge. Independentment de la *c:Caminada* a la que es faci la inscripció, l'*ES novaInscripció (c:Caminada)* és capturat pel controlador. Per tant si $gls:GLS$ té la responsabilitat d'emmagatzemar la *i:Inscripció*, llavors ha d'emmagatzemar les *Inscripcions* de *totes* les *Caminades*.

Destinació de l'enllaç dirigit d'emmagatzematge. L'emmagatzematge, per part del controlador, de *totes* les *Inscripcions* de *totes* les *Caminades* significa que l'enllaç dirigit d'emmagatzematge $gls:GLS \xrightarrow{atr} Inscripcions^*$ ha de tenir com a destinació les *Inscripcions* de *totes* les *Caminades* gestionades per $gls:GLS$.

Multiplicitat de la visibilitat d'emmagatzematge. Així, l'emmagatzematge exigeix que l'enllaç dirigit d'emmagatzematge $gls:GLS \xrightarrow{atr} Inscripció^*$ sigui *multiavaluat* i tingui com a destinació de l'enllaç *totes* les *Inscripcions* del sistema. Per tant, la visibilitat d'emmagatzematge $GLS \xrightarrow{atr} Inscripció^*$ és *multiavaluada*; i més concretament, cada enllaç dirigit que sigui realització d'aquesta visibilitat tindrà com a destinació *totes* les *Inscripcions* gestionades per l'objecte $gls:GLS$ origen de l'enllaç considerat.

Totes les Inscripcions juntes. Les necessitats de l'emmagatzematge impliquen que *totes* les *Inscripcions* del sistema s'accedeixen a través d'un mateix i únic enllaç dirigit d'emmagatzematge, realització de la visibilitat d'emmagatzematge $GLS \xrightarrow{atr} Inscripció^*$. És a dir, *totes* les *Inscripcions* estan agrupades en un mateix multiobjecte, destinació de la visibilitat d'emmagatzematge $GLS \xrightarrow{atr} Inscripció^*$!.

Repositori centralitzat. L’afirmació que totes les *Inscripcions* són accessibles a través d’un sol enllaç dirigit (l’enllaç dirigit d’emmagatzematge que té el controlador com a emmagatzemador) significa l’existència d’un *repositori centralitzat* de les *Inscripcions*, visible des del controlador.[→]

1.5.3 El multiobjecte com a objecte

Modisme MOMO sobre els multiobjectes: recordatori. Com hem vist,[→] un modisme habitual per tractar amb els multiobjectes amb llenguatges que no els suporten directament és el que rep el nom de *MOMO*. Aquest modisme exigeix que el disseny es preocupi de gestionar la ficció del multiobjecte com si d’un objecte més es tractés (amb l’excepció que es tracta d’un objecte que no és realització de cap component). Per tant s’ha d’estudiar qui en té visibilitat i quan i qui el crea. Per altra banda, continua sent responsabilitat de la implementació assegurar que els missatges rebuts per l’“objecte” multiobjecte es tractin com el disseny espera.

Aplicació de MOMO. En el cas que ens ocupa l’emmagatzematge exigeix un repositori centralitzat visible des del controlador. Per tant, l’enllaç dirigit d’emmagatzematge $gls:GLS \xrightarrow{atr} Inscripció^*$ és multiavaluat. En aquests casos MOMO exigeix tractar el multiobjecte destinació d’aquest enllaç dirigit com a un objecte, i en concret exigeix resoldre el problema de qui i quan crea aquest “objecte” multiobjecte.

Creador del multiobjecte d’Inscripcions. Qui ha creat el multiobjecte d’*Inscripcions*, del qual el controlador en té visibilitat d’atribut? Hem d’assumir que s’ha creat en algun moment anterior a l’execució d’aquest ES. Aquest moment no pot ser la realització d’un altre CU, perquè ningú no ens assegura que els agents externs es comportin exactament com volem. I tampoc pot ser un altre ES perquè l’especificació ens diu que *novaInscripcio(c:Caminada)* és el primer ES del CU *ferInscripcio*. L’únic punt de referència segur que tenim, en aquest cas, és la *inicialització del sistema*.

Sota el supòsit d’usar el controlador com a creador d’*i:Inscripció*, cal assumir que:

(21) ? !

- El controlador té visibilitat d’atribut d’un multiobjecte amb totes les *Inscripcions*
- En la inicialització del sistema és crea aquest multiobjecte

Excursió. (Creació primerenca) La idea subjacent de l’assumpció 21 és que com a PRE de l’ES *novaInscripcio(c:Caminada)* (vist com a ES del disseny)^{3,→} exigim l’existència del multiobjecte de totes les *Inscrip-*

³Per una explicació entre la diferència entre ES de l’especificació i del disseny, vegeu l’apartat 2.1.2.Lectura dels contractes dels ES, pàgina 22.

cions. Per assegurar aquesta PRE exigim la creació d'aquest multiobjecte en la inicialització del sistema.

Què passa però si suprimim aquesta PRE?. En aquest cas, abans de fer l'*add()* sobre el multiobjecte caldrà comprovar si aquest existeix; i en cas negatiu crear-lo.

En la proposta inicial hem fet una *creació primerenca* del multiobjecte: de fet en la inicialització del sistema no podem assegurar que en un futur es crearan Inscripcions! Reservem un espai, que potser no serà mai usat, a canvi de simplificació en el disseny.

En l'alternativa plantejada es crea l'espai només quan es necessita. El preu és que continuament estem preguntant si cal crear el multiobjecte. El disseny no és tant net; l'execució és més lenta; però som més conservadors amb l'espai usat

Generalment, i mentre no hi hagi raons fortes en contra, usarem la *creació primerenca*. És a dir, assumim que sempre que hem de fer un *add()* sobre un multiobjecte, hi ha una PRE que exigeix l'existència d'aquest multiobjecte. Això obliga a cercar un mecanisme per assegurar que la PRE sempre es compleix (en l'exemple, la creació en la inicialització del sistema).

1.5.4 Recuperació de la informació

Recuperació explícita de la Caminada. Ara és el moment d'analitzar les necessitats de recuperació de la *informació d'expertesa* de la creació de la *i:Inscripció*.⁴ En aquest cas, l'existència d'un *repositori centralitzat* d'Inscripcions exigeix una *recuperació explícita* de la *c:Caminada* a la que fa referència cada Inscripció. Tot seguit ho justifiquem.

Accés amb desconeixement. En el repositori (centralitzat) tenim barrejades totes les inscripcions, independentment del *a:c:Caminada* a la que es refereixen. La conseqüència és que quan hom accedeix a una *i:Inscripció* en desconeix la *c:Caminada* a la que *es refereix*.

Responsabilitat del manteniment de la Caminada. La possibilitat d'accedir a una *i:Inscripció* desconeixent-ne la *c:Caminada* a la que *es refereix* exigeix assignar a la *i:Inscripció* la responsabilitat de mantenir la *C:Caminada* a la que *es refereix*.⁵

Recuperació explícita de la Caminada. De tot plegat resulta que per poder recuperar la *c:Caminada* usada per a crear la *i:Inscripció* cal que aquesta la publiciti en la seva interfície: per exemple, *recuperarCaminada:<Caminada>* És a dir, la inscripció ha de tenir un mecanisme de recuperació explícita de la caminada

⁴A 10.11.Creació i recuperació de la informació, pàgina 191, s'expliquen els motius pels quals cal analitzar la recuperació de la informació.

⁵La idea, en termes planers, és la següent. Obtenim una *i:Inscripció* que pot referir-se a qualsevol *c:Caminada*, però no sabem a quina. Doncs res, li preguntem a la mateixa *i:Inscripció*!

Manteniment de la Caminada. La necessitat de poder recuperar explícitament la $c: \text{Caminada}$ a partir d'una $i: \text{Inscripció}$ significa que cal mantenir l'enllaç dirigit d'atribut $i: \text{Inscripció} \xrightarrow{\text{atr}} c: \text{Caminada}$. La visibilitat d'emmagatzematge corresponent a aquest enllaç dirigit, coincideix amb la modelització de l'associació "referent a".

Multiplicitat de la visibilitat de recuperació. L'associació "referent a" associa a cada *inscripció* una *camina*; per tant la visibilitat necessària per a la recuperació (que modelitza aquesta associació) ha de ser una *visibilitat d'atribut monoavaluada*: $\text{Inscripció} \xrightarrow{\text{atr}} \text{Caminada}$.

Creació de l'enllaç dirigit $i: \text{Inscripció} \xrightarrow{\text{atr}} c: \text{Caminada}$. Per tal de crear l'enllaç dirigit $i: \text{Inscripció} \xrightarrow{\text{atr}} c: \text{Caminada}$ cal, en el moment de crear la $i: \text{Inscripció}$, indicar a quina $c: \text{Caminada}$ es correspon la *Inscripció* que es vol crear: `novaInscripció(c: Caminada)`.

Excursió. (Opcionalitat de l'emmagatzematge) Estem considerant que no poden existir realitzacions d'*Inscripció* no enllaçades a cap realització de *Caminada*: l'associació *referent a* és obligada pel que fa a l'extrem de *Caminada*. En cas de permetre l'optativitat de l'associació, el missatge de creació no contindrà la $c: \text{Caminada}$, però caldrà dotar al component *Inscripció* d'un mecanisme per poder indicar quan calgui a quina realització de *Caminada* enllaçem la realització actual d'*Inscripció*. És a dir, cal dotar a *Inscripció* d'un mètode capaç de respondre al missatge `associaAmbCaminada(c: Caminada)`.

No cal una POST explícita. En el cas que ens ocupa una de les POST de l'ES exigeix explícitament que s'hagi creat un enllaç dirigit entre la $c: \text{Caminada}$ i la $i: \text{Inscripció}$.⁶ Però no sempre serà així. En general sempre cal analitzar la naturalesa de l'emmagatzematge i les possibilitats de recuperació de la informació necessitada per la creació.[→]

1.5.5 Anàlisi de l'acoblament de $M1$

El controlador com a mínim té visibilitat de paràmetre de la *Caminada*.⁷ A més, per ser-ne emmagatzemador, té visibilitat d'atribut d'*Inscripció*. I aquesta, al seu torn, ha de tenir visibilitat d'atribut de *Caminada* (ens cal per poder saber cada $i: \text{Inscripció}$ a quina $c: \text{Caminada}$ es refereix). Per mantenir aquesta darrera visibilitat cal, en la petició de creació de la *Inscripció* passar la *Caminada* a la que es refereix. Per tant tenim les visibilitats següents:

- Emmagatzematge: $\text{GLS} \xrightarrow{\text{atr}} \text{Inscripció}^*$

⁶El principi de la *Incorporació tardana* \rightarrow exigirà la supressió d'aquesta POST, que incorporarem en un altre ES. Tot i així l'anàlisi per trobar el creador de la $i: \text{Inscripció}$ és exactament el mateix. Arribem a la conclusió que *Caminada* és l'únic creador canònic; la diferència és que ara la darrera POST exigeix que l'emmagatzematge es faci de manera immediata, mentre que el principi de la *Incorporació tardana* demanarà posposar-lo per més endavant.

⁷Amb l'aplicació del principi de les *Connexions* i del principi de la *Cistella* $\rightarrow \cdot \rightarrow$ veurem que de fet la visibilitat és d'atribut.

- Recuperació: $\text{Inscripció} \xrightarrow{\text{atr}} \text{Caminada}$
- Arguments de l'ES: $\text{GLS} \xrightarrow{\text{par}} \text{Caminada}$
- Arguments del missatge de creació: $\text{Inscripció} \xrightarrow{\text{par}} \text{Caminada}$

1.6 M2: La c:Caminada com a creador

1.6.1 Aspecte general del model

Enllaços dirigits d'emmagatzematge. La $c:\text{Caminada}$ és creadora i emmagatzemadora d' $i:\text{Inscripció}$. L'emmagatzematge exigeix un enllaç dirigit (generalitzat) d'atribut entre la $c:\text{Caminada}$ argument de l'ES, i la nova $i:\text{Inscripció}$ creada: $c:\text{Caminada} \xrightarrow{\text{atr}} i:\text{Inscripció}^*$, tal que $i:\text{Inscripció}$ forma part de la destinació d'aquest enllaç dirigit.

Associació referent a. La visibilitat d'atribut $\text{Caminada} \xrightarrow{\text{atr}} \text{Inscripció}^*$ exigida per l'emmagatzematge és consistent amb les associacions de MC; en concret és la modelització de l'associació referent a entre *Caminada* i *Inscripció*, 1 – N.

1.6.2 Visibilitat d'emmagatzematge

Necessitats d'emmagatzematge. Independentment de la $i:\text{Inscripció}$ que es vagi a crear, la petició *novaInscripció(c:Caminada)* és capturada pel controlador, que delega la tasca de la creació de la nova $i:\text{Inscripció}$ a la $c:\text{Caminada}$, a través de l'enllaç dirigit de paràmetre $\text{gls}:\text{GLS} \xrightarrow{\text{par}} c:\text{Caminada}$. Així, cada $c:\text{Caminada}$ té la responsabilitat d'emmagatzemar tota $i:\text{Inscripció}$ que s'hi refereixi.

Destinació de l'enllaç dirigit d'emmagatzematge. L'emmagatzematge, per part de la *Caminada*, de totes les "seves" *Inscripcions* significa que l'enllaç dirigit d'emmagatzematge $c:\text{Caminada} \xrightarrow{\text{atr}} \text{Inscripcions}^*$ ha de tenir com a destinació el conjunt de totes les *Inscripcions* que es refereixen a la $c:\text{Caminada}$.

Multiplicitat de la visibilitat d'emmagatzematge. Així, l'emmagatzematge exigeix que l'enllaç dirigit d'emmagatzematge $c:\text{Caminada} \xrightarrow{\text{atr}} \text{Inscripció}^*$ sigui *multiavaluat* i tingui com a destinació de l'enllaç dirigit totes les *Inscripcions* de la $c:\text{Caminada}$. És a dir, la visibilitat d'emmagatzematge $\text{Caminada} \xrightarrow{\text{atr}} \text{Inscripció}^*$ és *multiavaluada*; i més concretament, cada enllaç dirigit que sigui realització d'aquesta visibilitat tindrà com a destinació totes les *Inscripcions* que es refereixen a la $c:\text{Caminada}$ origen de l'enllaç dirigit considerat.

Inscripcions agrupades per Caminada. Les necessitats de l'emmagatzematge impliquen que totes les *Inscripcions* sobre una mateixa $c:\text{Caminada}$ s'accedeixen a través d'un mateix i únic enllaç dirigit d'emmagatzematge, realització de la visibilitat d'emmagatzematge $\text{Caminada} \xrightarrow{\text{atr}} \text{Inscripció}^*$.

11.1 La *i:Inscripció* com a patata calenta

J.M. Merenciano

És a dir, totes les *Inscripcions* d'una mateixa *c:Caminada* estan agrupades en un mateix multiobjecte, que és la destinació de la visibilitat d'emmagatzematge $\text{Caminada} \xrightarrow{\text{atr}} \text{Inscripció}^*$!

Repositori distribuït per Caminades. La necessitat que les *Inscripcions* d'una *c:Caminada* siguin accessibles a través d'un sol enllaç dirigit (l'enllaç dirigit d'emmagatzematge que té la *c:Caminada* com a emmagatzemador) significa l'existència d'un *repositori distribuït*⁸ de les *Inscripcions*.⁸

Cada fragment del repositori és visible des de la *c:Caminada* pertinent; per tant el que tenim és un repositori de les *Inscripcions* *fragmentat per Caminades*.

9.4.Repositoris, pàgina 147

1.6.3 El multiobjecte com a objecte

Modisme MOMO sobre els multiobjectes: recordatori. Com hem vist,[→] un modisme habitual per tractar amb els multiobjectes amb llenguatges que no els suporten directament és el que rep el nom de *MOMO*. Aquest modisme exigeix que el disseny es preocupi de gestionar la ficció del multiobjecte com si d'un objecte més es tractés (amb l'excepció que es tracta d'un objecte que no és realització de cap component). Per tant s'ha d'estudiar qui en té visibilitat i quan i qui el crea. Per altra banda, continua sent responsabilitat de la implementació assegurar que els missatges rebuts per l'"objecte" multiobjecte es tractin com el disseny espera.

5.5.MOMO: Un modisme pels multiobjectes, pàgina 100

Aplicació de MOMO. En el cas que ens ocupa l'emmagatzematge exigeix un repositori fragmentat per *Caminades*. Per tant, l'enllaç dirigit d'emmagatzematge $\text{Caminada} \xrightarrow{\text{atr}} \text{Inscripció}^*$ és multiavaluat. En aquests casos *MO-MO* exigeix tractar el multiobjecte destinació d'aquest enllaç dirigit com a un objecte, i en concret exigeix resoldre el problema de qui i quan crea aquest "objecte" multiobjecte.

Creadors dels multiobjectes d'*Inscripcions*. Donada una *c:Caminada*, qui ha creat el multiobjecte d'*Inscripcions* sobre el qual té visibilitat d'atribut? Hem d'assumir que s'ha creat en algun moment anterior a l'execució d'aquest ES. Aquest moment no pot ser la realització d'un altre CU, perquè ningú no ens assegura que els agents externs es comportin exactament com volem. I tampoc pot ser un altre ES perquè l'especificació ens diu que *novaInscripcio(c:Caminada)* és el primer ES del CU *ferInscripcio*. Un punt de referència segur que tenim, en aquest cas, és la *creació de la c:Caminada*.

⁸Podem parlar de repositori distribuït perquè es compleixen les condicions següents:

- Tota *i:Inscripció* pertany a una *c:Caminada*
- Una mateixa *i:Inscripció* pertany a una única *c:Caminada*

La segona condició assegura que els diferents repositoris són disjunts. La primera condició ens diu que una *i:Inscripció* qualsevol pertany a la unió dels repositoris que mantenen les caminades.

Sota el supòsit d'usar la `c:Caminada` com a creador
d'`i:Inscripció`, cal assumir que:

- Cada `c:Caminada` té visibilitat d'atribut d'un multiobjecte amb totes les Inscripcions que s'hi refereixen
- En la creació de la `c:Caminada` es crea aquest multiobjecte

Excursió. (Creació primerenca) A l'excursió 1.5.3.Excursió. (Creació

1.5.3.Excursió. (Creació primerenca) [→] primerenca), s'ha introduït el concepte de la *creació primerenca*: independentment de si el sistema acabaria mantenint Inscripcions o no, inicialment es creava el multiobjecte responsable del seu emmagatzematge. Ara el cas és similar: independentment de si una `c:Caminada` concreta acabarà tenint o no Inscripcions, en crear la `c:Caminada` es crea també el multiobjecte responsable de l'emmagatzematge d'aquestes futures i hipotètiques Inscripcions.

1.6.4 Recuperació de la informació

Possibilitat de recuperació implícita de la Camianda. Ara és el moment d'analitzar les necessitats de recuperació de la *informació d'expertesa* de la creació de la *i*:Inscripció.^{9,→} En aquest cas, l'existència d'un repositori d'Inscripcions *fragmentat* per Caminades planteja els dubtes sobre la necessitat de recuperació de la *c*:Caminada a que fa referència cada Inscripció. Ho analitzem tot seguit.

10.11.Creació i recuperació de la informació,
pàgina 191

Accés amb coneixement. Si l'únic mecanisme que el sistema ofereix per accedir a les Inscripcions és a través de les Caminades, llavors si coneixem un *i*:Inscripció necessàriament prèviament cal conèixer la *c*:Caminada pertinent, la qual cosa fa innecessari qualsevol mecanisme de *recuperació explícita* de la *c*:Caminada a partir de la *i*:Inscripció.

Accés a les Inscripcions. El model que estem considerant emmagatzema les Inscripcions en un *repositori fragmentat per Caminades*. Per tal de recuperar dades emmagatzemades (les Inscripcions en aquest cas) cal accedir al repositori que l'emmagatzema (la *c*:Caminada a què *es refereix* la *i*:Inscripció).

Inscripcions compartides. Els diferents fragments del repositori d'Inscripcions no contenen *realitzacions compartides*,[→] però res no impedeix l'existència d'Inscripcions que siguin *realitzacions compartides* amb d'altres repositoris. (Per exemple, es pot tenir un repositori centralitzat amb les Inscripcions d'avui). En aquest cas, podríem accedir a algunes Inscripcions (les que són *realitzacions compartides*) a través de dos repositoris diferents.

9.4.Repositoris, pàgina 147

Ús d'un repositori alternatiu. En cas que totes o algunes de les Inscripcions estiguin en més d'un repositori, significa que es disposa d'un mecanisme d'accés alternatiu. I cal assegurar que sigui quin sigui el camí seguit per accedir a una *i*:Inscripció poguem accedir a la informació d'expertesa per a la creació d'aquesta *i*:Inscripció. La conseqüència és que davant de repositoris alternatius caldrà analitzar amb calma si calen mecanismes explícits de recuperació, o els mecanismes implícits són suficients.^{10,→}

??,?, pàgina ??

Simplicitat. De moment, però, el model considerat no exigeix aquests repositoris alternatius; i per tant podem assumir, en mor de la simplicitat, que l'únic mecanisme d'accés a les Inscripcions és a través de la *c*:Caminada pertinent. I per tant és innecessària la recuperació explícita de la *c*:Caminada des de la *i*:Inscripció.

⁹A 10.11.Creació i recuperació de la informació, pàgina 191 s'expliquen els motius pels quals cal analitzar la recuperació de la informació.

¹⁰Al capítol ??,?, pàgina ?? s'introdueixen els principis per resoldre aquesta problemàtica.

Excursió. (Opcionalitat de l'emmagatzematge) L'ús del repositori d'Inscripcions fragmentat per Caminades com a únic repositori d'Inscripcions, no és útil, com a model de l'associació *referent a* entre *Caminada* i *Inscripció*, si aquesta associació és *optativa* per l'extrem *Caminada*: no hi ha cap fragment del repositori d'Inscripcions que pugui emmagatzemar una *i : Inscripció* que no *es refereixi* a cap *c : Caminada*.

1.6.5 Anàlisi de l'acoblament de *M2*

Com en el model *M1*, el controlador com a mínim té visibilitat de paràmetre de la *Caminada*.¹¹ La *Caminada*, al seu torn, en ser emmagatzemadora d'Inscripció n'ha de tenir una visibilitat d'atribut. La *Inscripció*, però, no necessita cap visibilitat sobre ningú. Per tant tenim les visibilitats següents:

- Delegació: $GLS \rightarrow Caminada$
- Emmagatzematge: $Caminada \xrightarrow{atr} Inscripció^*$

17.2.3.Principi de les *Connexions*, pàgina 345

17.3.El flux comunicatiu intern, pàgina 348

¹¹ Amb l'aplicació del principi de les *Connexions* i del principi de la *Cistella* \rightarrow, \rightarrow veurem que de fet la visibilitat és d'atribut.

1.7 Comparativa dels dos possibles models

Responsable	Principi	Observacions
GLS	<i>Expert</i> <i>Creador orb</i> Viola: <i>Encarrilament</i> <i>Creador canònic</i>	<ul style="list-style-type: none"> • Repositori centralitzat d'Inscripcions • "referent a": $\text{Inscripció} \xrightarrow{atr} \text{Caminada}$ • Necessitat de recuperació de la Caminada • Recuperació: Cal passar la Caminada al mètode de creació <p>Cal una visibilitat $\text{GLS} \xrightarrow{atr} \text{Inscripció}^*$, que no es correspon a cap associació de MC</p>
c:Caminada	<i>Expert</i> <i>Baix acoblament</i> <i>Creador canònic</i>	<ul style="list-style-type: none"> • Repositori d'Inscripcions fragmentat per Caminades • "referent a": $\text{Caminada} \xrightarrow{atr} \text{Inscripció}^*$

Figura 11.1: Creadors d'i:Inscripció segons *Expert* i *Creador*

Anàlisi de l'acoblament. En el model *M1* el controlador està acoblat a Caminada i a Inscripció; en el model *M2* només està acoblat a Inscripció. A *M1* la Inscripció està acoblada a la Caminada (visibilitat $\text{I} \xrightarrow{atr} \text{C}$), mentre que a *M2* l'acoblament és a la inversa (visibilitat $\text{C} \xrightarrow{atr} \text{I}^*$). Per tant, sembla, que en termes d'acoblament és preferible el model *M2*.

Anàlisi de la cohesió. El model *M1* viola el principi de l'*Encarrilament*, que és un cas particular del principi d'*Alta cohesió*. Així, en termes de cohesió és preferible el model *M2*.

Visibilitat d'emmagatzematge. Ambdós models requereixen una visibilitat multiavaluada d'emmagatzematge sobre Inscripcions: a *M1* és una visibilitat des del controlador; a *M2* és una visibilitat des de Caminada. En el primer model tenim un *repositori centralitzat*; en el segon model Caminada és un *fragment* del repositori.

Modelització de l'associació "es refereix". A $M1$ l'associació "es refereix" entre *Caminada* i *Inscripció* es modelitza amb la *visibilitat de recuperació* $\text{Inscripció} \xrightarrow{\text{atr}} \text{Caminada}$; a $M2$ la modelització és a través de la *visibilitat d'emmagatzematge* $\text{Caminada} \xrightarrow{\text{atr}} \text{Inscripció}^*$ (que justament dirigeix els enllaços en el sentit oposat al proposat per la visibilitat de recuperació de $M1$!).

🔗 (23)

Prenem a $c:\text{Caminada}$ com a creador de la $i:\text{Inscripció}$.

1.8 Validesa de les decisions

La decisió de considerar com a preferible el model $M2$ ve avalada per l'anàlisi feta amb les dades actuals. I en concret tot es basa en els supòsits o hipòtesis següents, que caldrà validar al llarg de tot el disseny (segons el principi de la *Reconsideració*):

H (24)

1. En el MC no hi ha cap associació entre *GLS* i *Inscripció*
2. En cap moment necessitem un repositori centralitzat de les *Inscripcions*

Interpretació de les hipòtesis. La primera hipòtesi impedeix considerar el controlador com a *creador canònic*. La segona hipòtesi el que diu és que a les *Inscripcions* sempre hi accedirem des de la *Caminada* pertinent.

Escenaris alternatius. Res no impedeix tenir simultàniament un repositori centralitzat i un de descentralitzat. Com tampoc res impedeix tenir la doble visibilitat d'atribut $\text{Caminada} \leftrightarrow \text{Inscripció}$. En ambdós casos podem tenir problemes de consistència i de manteniment de les dades, però el disseny és possible.

Validesa condicionada del disseny. Cal tenir en compte que tota l'anàlisi feta per tal d'assignar responsabilitats recolza en les dues hipòtesis anteriors. Si les hipòtesis no es compleixen (com en el cas del doble repositori o de la doble visibilitat) llavors cap de les decisions d'assignació preses tenen validesa: *la teranyina s'esberla*.

2 POST: Enllaçar i i c

2.1 $M1$: el controlador com a creador de i

En el cas d'usar el creador no canònic d'Inscripció llavors el *controlador* emmagatzema totes Inscripcions. Com hem vist, per poder recuperar la informació de a quina c :Caminada es refereix cada i :Inscripció, cal mantenir una visibilitat d'atribut de i a c . I això ho aconseguim en donar la c :Caminada com a argument del mètode de creació de la i :Inscripció. El resultat no és res més que un enllaç dirigit $i \xrightarrow{atr} c$. Per tant la POST està assegurada, sense necessitat de fer res més.

2.2 $M2$: la c :Caminada com a creadora de i

En el cas d'usar el creador canònic d'Inscripció, cada Caminada emmagatzema les seves Inscripcions. Com hem vist, aquest emmagatzematge no és res més que un enllaç dirigit multiavaluat $c \xrightarrow{atr} i^*$. Per tant la POST està assegurada, sense necessitat de fer res més.

Capítol 12

Arguments implícits

1 Realització activa	220
1.1 Realització activa en una PRE	220
1.2 Realització activa en una POST	222
1.3 Realització activa en el MC i en el MComp	223
1.4 Realització activa en els contractes dels ES	224
2 Manteniment consistent dels contractes	224
2.1 Un problema d'inconsistència	224
2.2 Reconsideració del contracte de l'especificació	225
2.3 Anotacions sobre els contractes	227
3 Parèntesi sobre l'eficiència	227
3.1 Activació i eficiència	227
3.2 Perill de l'eficiència prematura	228
Principis	229
Definicions	229

Contingut detallat del capítol 12

1 Realització activa	220
Realitzacions actives en els contractes	220
1.1 Realització activa en una PRE	220
1.1.1 Visibilitat d'atribut	220
Arguments implícits	220
Visibilitat sobre la realització activa	220
Visibilitat local no possible	220
Necessitat de visibilitat d'atribut	220
Origen i destinació dels enllaços dirigits	220
1.1.2 Multiavaluació i activació	221
Problema i context	221
La problemàtica de la multiavaluació	221
Necessitat de la visibilitat d'atribut monoavaluada	221
Dues visibilitats	221
Multiavaluació de la visibilitat	221
1.1.3 Indirecció i activació	221
Comunicació i argument implícit	221
Possibilitat d'indirecció	222
1.2 Realització activa en una POST	222
Assegurar les visibilitats	222
Afegir una nova destinació a l'enllaç dirigit	222
1.3 Realització activa en el MC i en el MComp	223
1.3.1 Visibilitat d'activació i <i>Espill</i>	223
Visibilitat d'activació i MC	223
Coincidència de multiplicitats	223
Matis sobre la coincidència de visibilitats	223
Dificultat de la coincidència de multiplicitats	223
1.3.2 Visibilitat d'activació i els ES	223
No violació d' <i>Espill</i>	223
Una nova funció per la visibilitat d'atribut	223
Associacions implícites	223
Associacions implícites i <i>Espill</i>	224
1.4 Realització activa en els contractes dels ES	224
Contractes del disseny	224
Inconsistència entre disseny i especificació	224
Un problema general	224
2 Manteniment consistent dels contractes	224
2.1 Un problema d'inconsistència	224
Recuperació de la consistència entre els contractes	224
Anàlisi de les possibles solucions	224
2.2 Reconsideració del contracte de l'especificació	225
Una solució òbvia	225
Una especificació forçada	225

Descobriment inesperat	226
La solució òbvia no és satisfactòria	226
Una pregunta amb resposta	226
2.3 Anotacions sobre els contractes	227
El problema	227
Propietats a mantenir	227
La solució	227
Estabilitat de l'especificació	227
3 Parèntesi sobre l'eficiència	227
3.1 Activació i eficiència	227
Activació i indirecció	227
Ús de la indirecció	227
Ineficiència en la indirecció	228
3.2 Perill de l'eficiència prematura	228
Eficiència en segon pla	228
Eficiència prematura	228
Primer disseny, després eficiència	228
L'eficiència com a mesura de contrast	228
Principis	229
Definicions	229

1 Realització activa

Realitzacions actives en els contractes. En els contractes és molt habitual demanar (PRE) o assegurar (POST) l'existència d'una *realització activa*. Per això cal estudiar amb deteniment la semàntica associada.

1.1 Realització activa en una PRE

1.1.1 Visibilitat d'atribut

Arguments implícits. En una PRE l'exigència d'una realització activa es correspon a l'existència d'un *argument per omissió o implícit*: l'operació que respon el missatge actúa sobre la realització activa que exigeix la PRE.

Visibilitat sobre la realització activa. Així l'exigència d'una realització activa en la PRE d'un missatge $m()$ significa que el receptor del missatge ha de tenir accés, com a mínim mentre està responent el missatge $m()$, a la realització activa exigida per la PRE de $m()$. És a dir, el receptor d'un missatge $m()$ ha de tenir *visibilitat* sobre les realitzacions actives exigides per la PRE de $m()$, i això sense necessitat de rebre-les com a arguments de $m()$.

Visibilitat local no possible. La visibilitat entre el receptor d'un missatge i les realitzacions actives exigides per la PRE del missatge en qüestió no pot ser local, ja que la localitat exigeix la creació de l'enllaç dirigit dins del context d'emissió de l'operació considerada. Per tant la PRE, que demana l'existència *abans* d'enviar el missatge capturat per l'operació, no podrà ser mai assolida.

Necessitat de visibilitat d'atribut. Un cop descartada la visibilitat local, l'exigència que durant la realització de l'operació que respon el missatge $m()$ el receptor del missatge ha de tenir visibilitat sobre la realització activa demanada per la PRE de $m()$, necessàriament significa que aquesta visibilitat ha de ser d'atribut.

(33)

Necessàriament tot objecte que rep un missatge ha de tenir visibilitat d'atribut sobre les realitzacions actives exigides per la PRE del missatge en qüestió

Origen i destinació dels enllaços dirigits. La visibilitat d'atribut necessària per mantenir una *realització activa* té el seu *origen* en l'objecte que rep el missatge que ha de treballar amb la realització activa com a paràmetre implícit. La *destinació* de l'enllaç dirigit en qüestió és justament la realització activa que usem com a argument implícit.

Exemple 60 (Exigir una realització activa) *Suposem que el component Caminada ofereix una operació `nouParticipant(nom)` que, entre d'altres, té una PRE que demana l'existència d'una*

12.1 Realització activa

J.M. Merenciano

$i : \text{Inscripció activa}$.

Això significa l'existència d'una visibilitat *Caminada* $\xrightarrow{\text{atr}}$ *Inscripció*. Quan una $c : \text{Caminada}$ concreta rebi el missatge `nouParticipant(nom)` la PRE exigeix l'existència, per a alguna $i : \text{Inscripció}$, d'un enllaç dirigit $c : \text{Caminada} \xrightarrow{\text{atr}} i : \text{Inscripció}$.

1.1.2 Multiavaluació i activació

Problema i context. Suposem que el contracte del mètode $m()$ del component C exigeix l'existència d'una realització activa $a : A$. I suposem l'existència d'una visibilitat multiavaluada $C \xrightarrow{\text{atr}} A^*$.

Què vol dir que tenim una *realització activa* $a : A$?

(25) ?

La problemàtica de la multiavaluació. La visibilitat d'atribut multiavaluada $C \xrightarrow{\text{atr}} A^*$ implica que donat un objecte $c : C$ a través d'ell podem accedir a molts objectes diferents, tots ells realització del component A . Quin és però, d'entre tots aquests objectes, l'argument implícit que cal considerar pel mètode $m()$?

Necessitat de la visibilitat d'atribut monoavaluada. Per tal d'eliminar l'ambigüitat que representa la multiavaluació, cal tenir *visibilitat d'atribut monoavaluada* sobre l'objecte que cal considerar com a implícit.

Dues visibilitats. Tot plegat vol dir que, a part de la visibilitat d'atribut multiavaluada $C \xrightarrow{\text{atr}} A^*$ exigida pel disseny pels motius que sigui, l'exigència d'un $a : A$ actiu significa que també cal mantenir una visibilitat d'atribut *monoavaluada* $C \xrightarrow{\text{atr}} A$ que indiqui, d'entre tots els objectes que són destinació de l'enllaç dirigit $c : C \xrightarrow{\text{atr}} A^*$, quin és l'objecte $a : A$ que cal considerar com a objecte implícit.

Multiavaluació de la visibilitat. Els paràgrafs anteriors justifiquen que si la PRE exigeix una realització activa la visibilitat d'atribut responsable de mantenir activa alguna realització serà monoavaluada. Ara bé, si la PRE exigeix diverses realitzacions actives, podem mantenir aquestes realitzacions amb múltiples visibilitats (monoavaluades), o amb una única visibilitat multiavaluada.

1.1.3 Indirecció i activació

Comunicació i argument implícit. L'exigència d'una *realització activa* és l'exigència d'un *enllaç dirigit d'atribut* que permeti usar la destinació d'aquest enllaç com a argument implícit de les peticions fetes sobre l'objecte origen de

l'enllaç. És a dir, el que exigeix la PRE és que qui rep la petició tingui coneixement d'una determinat objecte; i aquest coneixement hem vist que cal plasmar-lo en forma de visibilitat d'atribut monoavaluada.¹

Possibilitat d'indirecció. L'exigència d'una *realització activa* no imposa cap condició en quin és el mecanisme per recuperar aquesta realització per a usar-la com a argument implícit. Són els mecanismes de visibilitat del disseny els qui imposen la necessitat d'una visibilitat d'atribut. Però res impedeix que aquesta sigui una *visibilitat indirecta*.

Definició. Realització activa

Una realització activa és una realització que compleix les següents propietats:²

- És un argument implícit sobre el que actua el missatge
- És una realització d'un component (objecte) sobre el que el component receptor d'un missatge en té visibilitat (monoavaluada) d'atribut indirecta

Definició. Visibilitat d'activació

Anomenem visibilitat d'activació la visibilitat d'atribut generalitzada, directa o indirecta, usada per mantenir realitzacions actives.

(34)

La multiplicitat de la *visibilitat d'activació* ha de ser consistent amb el nombre de realitzacions actives exigides.

1.2 Realització activa en una POST

Assegurar les visibilitats. En una POST l'asseveració de l'existència d'una realització activa el que diu és que s'han establert els mecanismes corresponents per tal que aquesta realització pugui considerar-se com a argument per omissió allà on convingui (és a dir, allà on ens ho digui el contracte del missatge). En concret, ens diu que s'han establert les *visibilitats d'activació* necessàries.

Afegir una nova destinació a l'enllaç dirigit. Hem vist que en la PRE l'exigència d'una *realització activa* es tradueix en una visibilitat d'atribut (la *visibilitat d'activació*). Suposem que aquesta visibilitat és $A \xrightarrow{atr} B^*$. Llavors, per assegurar

¹En el cas que la PRE demani el coneixement de més d'un objecte, aquest coneixement pot expressar-se en termes de visibilitat d'atribut multiavaluada.

²Ambdues propietats són equivalents. La primera focalitza en l'especificació dels missatges; la segona, en el disseny de l'operació que respon al missatge.

12.1 Realització activa

J.M. Merenciano

que determinada realització $act : B$ es manté activa cal assegurar que $act : B$ és una³ de les destinacions de l'enllaç dirigit d'atribut corresponent, realització de la visibilitat d'activació. Per tant n'hi ha prou en afegir $act : B$ com a nova destinació en l'enllaç dirigit pertinent.

1.3 Realització activa en el MC i en el MComp

1.3.1 Visibilitat d'activació i *Espill*

Visibilitat d'activació i MC. La *visibilitat d'activació* necessària per modelitzar l'existència d'una realització activa, per ser una visibilitat d'atribut, també modelitza una possible associació del MC. Però no sempre aquesta associació realment apareix en el MC; i llavors estem davant una violació del principi de l'*Espill*.

Coincidència de multiplicitats. Suposem una *visibilitat d'activació* $A \xrightarrow{atr} B^*$. Si en a MC existeixi una associació entre A i B amb exactament la mateixa multiplicitat que l'exigida per la *visibilitat d'activació*, llavors la pròpia *visibilitat d'activació* modelitza aquesta associació.

Matís sobre la coincidència de visibilitats. Cal tenir present, però, que la coincidència de multiplicitats demanada en el paràgraf anterior ha de ser una coincidència exacta, no d'ordre de magnitud. És a dir, per cada objecte $a : A$, les realitzacions de B actives que ha de mantenir la visibilitat d'activació són justament totes les realitzacions del concepte B que estan associades al concepte concret, realització de A , que modelitzem amb l'objecte $a : A$.

Dificultat de la coincidència de multiplicitats. El matís acabat d'expressar sobre la coincidència de multiplicitats converteix en impràctic aquest criteri. Per tant cal assumir que *en general la visibilitat d'activació no modelitza les associacions de MC*.

1.3.2 Visibilitat d'activació i els ES

No violació d'*Espill*. ;algrat tot, les visibilitats d'activació exigides pels contractes dels ES podem considerar que *no* violen el principi de l'*Espill*. Vegem-ho.

Una nova funció per la visibilitat d'atribut. En els contractes de l'especificació demanem una realització activa, però enlloc expressem com ni qui gestiona les realitzacions actives. El que se'ns diu és que ha d'existir tal mecanisme. La *Modelització continua* ens diu que aquest mecanisme ha d'existir d'alguna manera a MComp. L'anàlisi feta més amunt el que ens diu que el mecanisme de MComp per modelitzar l'existència de realitzacions actives és mitjançant visibilitats d'atribut. En conseqüència podem considerar que la *visibilitat d'atribut* té una funció addicional que és la de modelitzar les peticions de realitzacions actives dels contractes dels ES; i per tant no es viola *Espill*.

³En el cas de visibilitats monoavaluades, aquest "una" és un "la".

Asociacions implícites. Una visió alternativa és considerar que la petició d'una realització activa en els contractes dels ES és una manera alternativa d'exigir una associació especial entre el sistema (que és qui rep la petició en forma d'ES) i el concepte de qui les realitzacions actives en són realització. És a dir, considerem que a MC hi ha associacions dibuixades explícitament, i *associacions implícites* amagades en la idea de les realitzacions actives, i que tenen un dels seus extrems en el propi sistema o problema i no pas en un dels conceptes d'aquest.

Associacions implícites i Espill. Si considerem l'exigència de les realitzacions actives com l'existència d'*associacions implícites*, llavors tenim que les *visibilitat d'activació* modelitzen precisament aquestes associacions implícites i per tant no es viola *Espill*.

1.4 Realització activa en els contractes dels ES

Contractes del disseny. Qui respon als ES són els controladors. Per tant una realització activa en el contracte del disseny d'un ES^{4,→} significa un enllaç dirigit d'atribut (directe o indirecte), realització de la *visibilitat d'activació*, tal que el seu origen és el controlador, i una de les seves destinacions és la realització que es vol activa.⁵

Inconsistència entre disseny i especificació. Els canvis exigits pel manteniment de les realitzacions actives en els contractes del disseny converteixen aquests contractes en inconsistents respecte els contractes de l'especificació. Ara ja no tenim un contracte amb dues lectures, tenim dos contractes diferents.

Un problema general. Aquest és un problema amb el que ens trobarem sovint. I no només davant d'una visibilitat d'activació. Per això li dediquem tot el següent apartat.

2 Manteniment consistent dels contractes

2.1 Un problema d'inconsistència

Recuperació de la consistència entre els contractes. Quan una decisió de disseny (com el manteniment de la visibilitat d'activació) demana una modificació dels contractes del disseny tenim dues opcions:

- Modifiquem el contracte de l'especificació per fer-lo consistent amb el nou contracte del disseny
- No modifiquem el contracte del disseny

⁴A 2.1.2.Lectura dels contractes dels ES, pàgina 22 s'analiza la diferència entre els contractes del disseny i els contractes de l'especificació.

⁵En el cas de mantenir una sola realització activa, la visibilitat d'activació és monoavaluada, i per tant l'enllaç dirigit d'atribut que cal mantenir és aquell que va del controlador a l'objecte actiu. Si volem mantenir moltes realitzacions actives, l'origen de l'enllaç dirigit d'activació és el controlador, i tots els objectes que són destinació d'aquest enllaç dirigit són els objectes actius.

Anàlisi de les possibles solucions. Tot seguit analitzem les dues propostes de solució al problema de la inconsistència.

2.2 Reconsideració del contracte de l'especificació

Una solució òbvia. Pels motius que sigui, ens veiem obligats a modificar un contracte del disseny. En aquests moments una solució òbvia per mantenir la consistència dels contractes, és reconsiderar el corresponent contracte de l'especificació.

Exemple 61 (Reconsideració del contracte de l'especificació)

Sigui el CU `ferInscripcio` segons el model M2: el creador d'`Inscripció` és `Caminada`. I sigui la visibilitat indirecta d'activació $K \rightarrow \text{Caminada} \rightarrow \text{Inscripció}$.⁶→

Això significa que en el **disseny** necessitem les següents POST (mostrem només les assercions que es corresponen a l'activació), que assegurin que el controlador manté una emmagatzematge monoavaluat de la `:Caminada` sobre la que s'està fent la `:Inscripció`, i que aquesta `:Caminada` manté una emmagatzematge monoavaluat de la `:Inscripció` que estem creant; i que ambdós emmagatzematges són locals al CU.

ES	POST
<code>novaInscripcio(c:Caminada)</code>	$:K \rightarrow c:Caminada$ $c:Caminada \rightarrow i:Inscripció$
<code>fiInscripcio()</code>	$:K \rightarrow \text{null}$ $c:Caminada \rightarrow \text{null}$

Com aquests canvis es poden reflectir en l'especificació? Una manera és considerar que en l'especificació parlar d'una realització activa significa tenir-ne un coneixement directe. Llavors podríem modificar els contractes de la següent manera (marquem amb **blau** els canvis):

ES	POST
<code>novaInscripcio(c:Caminada)</code>	<ul style="list-style-type: none">• La <code>c:Caminada</code> està activa• <code>i:Inscripció</code> està activa per a la <code>c:Caminada</code>
<code>fiInscripcio()</code>	<ul style="list-style-type: none">• No hi ha cap caminada activa• No hi ha cap inscripció activa per a cap caminada

En aquest cas la reconsideració dels contractes de l'especificació no és massa problemàtica, tot i que obliga a fixar la semàntica de l'activació en el sentit que en l'especificació només entenem l'activació directa.

13. ^{GLS} POST: La `i:Inscripció` es manté activa, pàgina 231

⁶El capítol 13. ^{GLS} POST: La `i:Inscripció` es manté activa, pàgina 231 analitza detalladament l'activació en el cas de *Gastem la sola*.

Una especificació forçada. La reconsideració dels contractes de l'especificació, però, pot portar a contractes massa forçats. En concret podem fer aparèixer en l'especificació condicions que en aquest moment o no tenen significat o no s'entén el perquè de la seva necessitat. El resultat és una contracte farragós i massa complex.

Exemple 62 (Especificació forçada) *Què significa que la i:inscripció està activa per a la c:Caminada?*

En l'especificació les associacions són simètriques, i no existeix la idea d'accés o visibilitat. L'assertió anterior, per tant, és molt difícil d'entendre.

Segurament el que volem dir és que la i:Inscripció diferenciada (activa) depèn de la c:Caminada diferenciada que considerem. Però fins i tot si explicitem clarament la semàntica de l'assertió aquesta apareix forçada.

De cara a construir una i:Inscripció, comencem per indicar per quina c:Caminada és, i tot seguit introduïm els diferents participants. Per tant, la semàntica del sistema exigeix deixar una i:Inscripció activa. Demanar també la c:Caminada i relativitzar l'activació de la i:Inscripció a l'activació de la c:Caminada és exagerat: complica l'especificació que no té cap necessitat d'aquests afegits.

Descobriment inesperat. En moltes ocasions el fet de reconsiderar l'especificació davant de certes exigències del disseny porta a una especificació sobrecarregada i molt forçada. En d'altres ocasions, però, les exigències del disseny es poden fer adonar d'oblits o mancances en l'especificació; i en aquest cas la reconsideració és necessària.

(35)

Reconsideració dels contractes de l'especificació.

Les exigències de disseny poden donar nova llum a la comprensió del problema. En aquest cas la reconsideració de l'especificació és necessària. En qualsevol altre cas la reconsideració el que fa és malmetre l'especificació.

La solució òbvia no és satisfactòria. Volem una especificació simple i clara. Per tant reconsideracions com les de l'exemple 61 → s'han d'evitar. Però si no fem la reconsideració mantenim dos contractes diferents, el de l'especificació i el disseny.

Exemple 61.Reconsideració del contracte de l'especificació, pàgina 225

? (26)

Com podem mantenir un sol contracte, mb dues lectures, i tot i així incloure d'alguna manera les exigències del disseny?

Una pregunta amb resposta. En el següent apartat responem aquesta pregunta.

2.3 Anotacions sobre els contractes

El problema. Determinades exigències del disseny ens obliguen a modificar algun dels contractes del disseny. Però si reconsiderem l'especificació a partir d'aquests nous contractes, obtenim una especificació farragosa i sobre-garregada, i ho volem evitar.

Propietats a mantenir. El que ens interessa és mantenir un únic contracte, llegible en termes del disseny i de l'especificació; que aquest contracte sigui simple; i que quan el llegim en termes del disseny hi puguem llegir totes les decisions i exigències preses,

La solució. La solució que proposem és la de mantenir el contracte, però afegir-hi anotacions que seran interpretables en el disseny.

Exemple 63 (Contracte anotat) Tornem a l'exemple 61. → El contracte el mantenim com l'original:

ES	POST
novaInscripcio(c:Caminada)	La i:Inscripció està activa
fiInscripcio()	No hi ha cap inscripció activa

I ara n'hi ha prou en afegir la següent anotació:

La i:Inscripció es manté activa amb la visibilitat d'indirecció indirecta $K \rightarrow \text{Caminada} \rightarrow \text{Inscripcio}$

Exemple 61.Reconsideració del contracte de l'especificació, pàgina 225

Estabilitat de l'especificació. L'especificació ha de ser el més estable possible.

Això significa que les exigències que pugui posar el disseny en els contractes només es reflectiran en l'especificació si ha estat un oblit d'aquest; altrament aquestes exigències s'introduiran com a *anotacions* sobre els contractes, que no es modificaran.

Principi



3 Parèntesi sobre l'eficiència

3.1 Activació i eficiència

Activació i indirecció. En el cas de tenir una *visibilitat d'activació indirecta* l'accés a la realització activa s'haurà de fer sempre a través d'un enllaç dirigit indirecte.

Ús de la indirecció. Per accedir a un objecte a través d'un enllaç dirigit indirecte s'ha de passar per tots els intermediaris d'aquest enllaç: la petició d'accés s'ha de delegar a través de la cadena d'indirecció. Però cada delegació requereix una nova petició; i això és temps.

Ineficiència en la indirecció. Per tant, l'ús d'una *visibilitat d'activació indirecta* significa *ineficiència en el temps*.^{7, →}

13.4. Anàlisi de l'eficiència, pàgina 242

? (27)

En dissenyar, cal considerar la *visibilitat d'activació indirecta*, quan sabem que la indirecció que implica aquesta és un factor d'*ineficiència*?

Aquesta pregunta la resol el principi de l'*Eficiència en segon pla*.

3.2 Perill de l'eficiència prematura

Principi



Eficiència en segon pla. En el moment de dissenyar, les qüestions d'eficiència s'han de mantenir en un segon pla.

Eficiència prematura. Un dels errors més típics en dissenyar és caure en l'*eficiència prematura*: estem tant preocupats per ser eficients que ens oblidem de fer les coses bé.

Primer disseny, després eficiència. El procedir correcte és el d'aconseguir, en primer lloc, un disseny consistent; després ens podrem preocupar d'aspectes com l'eficiència.

L'eficiència com a mesura de contrast. El principi de l'*Eficiència en segon pla* diu que no hem d'oblidar les qüestions d'eficiència, però que només les hem de tenir en compte, *de moment*, quan haguem d'escollir entre dues alternatives que els principis de disseny ens ofereixen com igualment vàlides. L'eficiència ens pot servir com a mesura de contrast per decantar-nos cap a una o altra alternativa. El que no s'hi val, però, és amb l'excusa de l'eficiència violar tots els principis de disseny.

⁷Com veurem a 13.4. Anàlisi de l'eficiència, pàgina 242, en alguns casos també pot ser *ineficiència d'espai*.

Principis i definicions del capítol

Principis

Eficiència en segon pla, 228
Estabilitat de l'especificació,
227

Definicions

Realització activa, 222
Visibilitat d'activació, 222

Capítol 13

GLS POST: La i : Inscripció es manté activa

1	Activació de la i : Inscripció	235
2	Activació a $M1$	235
2.1	Anàlisi prèvia de $M1$	235
2.2	Activació per visibilitat directa a $M1$	235
2.3	Activació per visibilitat indirecta a $M1$	236
2.4	Comparativa de les dues possibilitats sobre $M1$	237
3	Activació a $M2$	239
3.1	Anàlisi prèvia de $M2$	239
3.2	Activació per visibilitat directa a $M2$	239
3.3	Activació per visibilitat indirecta a $M2$	240
3.4	Comparativa de les dues possibilitats sobre $M2$	241
4	Anàlisi de l'eficiència	242
4.1	Una solució poc eficient	242
4.2	Disseny sense anàlisi exhaustiva	242
4.3	Anem a violar l' <i>Eficiència en segon pla</i>	243
4.4	Què fem davant la manca d'eficiència	245
5	Conclusions i decisions de disseny	245
6	Els models en la situació actual	247

Contingut detallat del capítol 13

1 Activació de la i:Inscripció	235
Semàntica de l'activació	235
Dos models per analitzar	235
2 Activació a M1	235
2.1 Anàlisi prèvia de M1	235
Recordatori del model considerat M1	235
Visibilitats actuals del model M1	235
Visibilitats actuals sobre Inscripció	235
Dues possibilitats	235
2.2 Activació per visibilitat directa a M1	235
Visibilitat d'activació de la i:Inscripció	235
Manteniment de l'activació directa a M1	235
Visibilitats de M1 amb activació amb <i>visibilitat directa</i>	236
Una dependència transformada	236
2.3 Activació per visibilitat indirecta a M1	236
Visibilitat d'activació de la i:Inscripció	236
D'on surt la visibilitat indirecta?	236
Manteniment de l'activació indirecta a M1	236
Visibilitats de M1 amb activació amb <i>visibilitat indirecta</i>	237
2.4 Comparativa de les dues possibilitats sobre M1	237
Anàlisi subjectiva	237
Aproximacions quantitatives	237
Anàlisi de l'acoblament	237
Dependències i nous acoblaments.	237
Excursió. (Acoblament i dependències)	237
Anàlisi <i>quantitativa</i> de l'acoblament.	238
Anàlisi <i>qualitativa</i> de l'acoblament.	238
Anàlisi superficial de la cohesió	238
Anàlisi afinada de la cohesió.	238
Anàlisi <i>quantitativa</i> de la cohesió.	238
Anàlisi <i>qualitativa</i> de la cohesió.	238
Resultat de l'anàlisi de les propostes a M1	238
Anàlisi interrompuda	239
3 Activació a M2	239
3.1 Anàlisi prèvia de M2	239
Recordatori del model considerat M2	239
Visibilitats actuals del model M2	239
Visibilitats actuals sobre Inscripció	239
Dues possibilitats	239
3.2 Activació per visibilitat directa a M2	239
Visibilitat directa d'activació de la i:Inscripció	239
Manteniment de l'activació directa a M2	239
Visibilitats de M2 amb activació amb <i>visibilitat directa</i>	240

	Una dependència transformada	240
	Violació d' <i>Espill</i>	240
3.3	Activació per visibilitat indirecta a <i>M2</i>	240
	Visibilitat d'activació de la <i>i</i> :Inscripció	240
	Introducció de la Caminada activa	240
	Activació de les Inscripcions local a cada Caminada	240
	Visibilitats de <i>M2</i> amb activació amb <i>visibilitat indirecta</i>	241
	Una dependència transformada	241
3.4	Comparativa de les dues possibilitats sobre <i>M2</i>	241
	Anàlisi quantitativa i anàlisi qualitativa	241
	Anàlisi de l'acoblament	241
	Anàlisi de la cohesió	241
	Resultat de l'anàlisi de les propostes a <i>M2</i>	241
4	Anàlisi de l'eficiència	242
4.1	Una solució poc eficient	242
	Problemes d'eficiència	242
	<i>Ineficiència en espai</i>	242
	Situació hipotètica.	242
	<i>Ineficiència en temps</i>	242
	Volem ser eficients	242
	Eficiència en segon pla	242
4.2	Disseny sense anàlisi exhaustiva	242
	El nostre camí	242
	El camí habitual	242
	El camí habitual en el nostre cas	243
	Una assignació problemàtica	243
	Asseveració falsa	243
	Una hipòtesi de treball	243
4.3	Anem a violar l' <i>Eficiència en segon pla</i>	243
	Millorar l'eficiència en temps	243
	De l'eficiència en temps al desastre	243
	Canvis en el context de l'anàlisi	244
	Visibilitats actuals sobre <i>Inscripció</i>	244
	Creador orb i creador canònic	244
	Canvis en les condicions d'anàlisi	244
	Reconsideració	244
	Resultat de la consideració	244
	Conseqüències de l'intent de millora de l'eficiència en temps	244
	Compte no caiguem a la trampa	244
	Millora de l'eficiència en espai	245
4.4	Què fem davant la manca d'eficiència	245
	Teranyina	245
	Documentació de l'eficiència	245
5	Conclusions i decisions de disseny	245
	Resum de l'anàlisi	245
	Conclusió i decisió de disseny	246

Intermediari de la visibilitat	246
Canvis en els contractes	246
6 Els models en la situació actual	247
Creació de la i:Inscripció	247
Activació de la i:Inscripció	248

1 Activació de la *i:Inscripció*

Què vol dir que la *i:Inscripció* esdevé activa?

(28) ?

Semàntica de l'activació. Segons el que acabem de veure, \rightarrow per mantenir una *Inscripció* activa cal una visibilitat monoavaluada d'atribut (in)directa des del controlador.

12. Arguments implícits, pàgina 217

Dos models per analitzar. Ja tenim força criteris per preferir el model *M2* per sobre del model *M1*. Malgrat tot, per completesa expositiva, analitzarem ambdós models.

2 Activació a *M1*

2.1 Anàlisi prèvia de *M1*

Recordatori del model considerat *M1*. *gls:GLS* té els rol de controlador, i de creador i emmagatzemador de la *i:Inscripció*. La *c:Caminada* no té cap rol assignat.

Visibilitats actuals del model *M1*. En el model que estem considerant tenim les següents visibilitats:

$GLS \xrightarrow{par} Caminada$	Argument ES
$GLS \xrightarrow{loc} Inscripció$	Creació
$GLS \xrightarrow{atr} Inscripció^*$	Emmagatzematge
$Inscripció \xrightarrow{atr} Caminada$	Recuperació
$Inscripció \xrightarrow{par} Caminada$	Manteniment de la recuperació

Visibilitats actuals sobre *Inscripció*. Qui té visibilitat sobre la *i:Inscripció* és *gls:GLS*.

Dues possibilitats. Per tant per mantenir la *i:Inscripció* activa tenim dues possibilitats:¹

- *Visibilitat directa* $GLS \xrightarrow{atr} Inscripció$
- *Visibilitat indirecta* $GLS \xrightarrow{atr} Caminada \xrightarrow{atr} Inscripció$

2.2 Activació per visibilitat directa a *M1*

Visibilitat d'activació de la *i:Inscripció*. Com a visibilitat d'activació usem $GLS \xrightarrow{atr} Inscripció$.

¹La justificació d'aquestes possibilitats es realitza tot seguit dins l'anàlisi de cadascuna d'elles.

Manteniment de l'activació directa a $M1$. El controlador crea la $i : \text{Inscripció}$; i un cop creada se l'emmagatzema de manera monoavaluada. Com a conseqüència, la visibilitat $\text{GLS} \rightarrow \text{Inscripció}$ que teníem com a dependència local, es converteix en una dependència d'atribut.

Visibilitats de $M1$ amb activació amb visibilitat directa. En el model que estem considerant tenim les següents visibilitats:

$\text{GLS}^{\text{par}} \rightarrow \text{Caminada}$	Argument ES
$\text{GLS}^{\text{loc}} \rightarrow \text{Inscripció}$	Creació
$\text{GLS}^{\text{atr}} \rightarrow \text{Inscripció}^*$	Emmagatzematge
$\text{Inscripció}^{\text{atr}} \rightarrow \text{Caminada}$	Recuperació
$\text{Inscripció}^{\text{par}} \rightarrow \text{Caminada}$	Manteniment de la recuperació
$\text{GLS}^{\text{atr}} \rightarrow \text{Inscripció}$	Creació i activació

Una dependència transformada. La dependència local resultat de la creació de la $i : \text{Inscripció}$ es transforma en una dependència d'atribut per tal de mantenir emmagatzemada la $i : \text{Inscripció}$ acabada de crear. Per això hem suprimit la dependència local i n'hem fet aparèixer una d'atribut.

2.3 Activació per visibilitat indirecta a $M1$

Visibilitat d'activació de la $i : \text{Inscripció}$. Com a visibilitat d'activació usem la visibilitat indirecta $\text{GLS}^{\text{atr}} \rightarrow \text{Caminada} \rightarrow \text{Inscripció}$. Tot seguit analitzem com hem arribat a aquesta visibilitat, i quines implicacions té.

D'on surt la visibilitat indirecta? L'única dependència incident a $i : \text{Inscripció}$ és $\text{GLS} \rightarrow \text{Inscripció}$. Si la usem per a l'activació estem en el cas d'activació per visibilitat directa. Però volem una visibilitat d'activació indirecta. L'única possibilitat, sense afegir nous components a $MComp$, és $\text{GLS}^{\text{atr}} \rightarrow \text{Caminada}^{\text{atr}} \rightarrow \text{Inscripció}$. El resultat és una cadena de visibilitats que del controlador arriba a $i : \text{Inscripció}$ passant per $c : \text{Caminada}$.

Manteniment de l'activació indirecta a $M1$. Per mantenir l'activació de la $i : \text{Inscripció}$, el controlador, un cop l'ha creada, l'ha de comunicar a la $c : \text{Caminada}$ per tal que l'emmagatzemi de manera monoavaluada. I a més el controlador també ha d'emmagatzemar la $c : \text{Caminada}$ de manera monoavaluada.

Visibilitats de *M1* amb activació amb visibilitat indirecta.

$GLS \xrightarrow{par} \text{Caminada}$	Argument ES
$GLS \xrightarrow{loc} \text{Inscripció}$	Creació
$GLS \xrightarrow{atr} \text{Inscripció}^*$	Emmagatzematge
$\text{Inscripció} \xrightarrow{atr} \text{Caminada}$	Recuperació
$\text{Inscripció} \xrightarrow{par} \text{Caminada}$	Manteniment de la recuperació
$GLS \xrightarrow{atr} \text{Caminada}$	Activació
$\text{Caminada} \xrightarrow{atr} \text{Inscripció}$	Activació
$\text{Caminada} \xrightarrow{par} \text{Inscripció}$	Manteniment de l'activació

2.4 Comparativa de les dues possibilitats sobre *M1*

Anàlisi subjectiva. Tant l'anàlisi de la cohesió \rightarrow com la de d'acoblament \rightarrow és una anàlisi subjectiva, ja que la magnitud que s'ha de maximitzar o minimitzar es defineix en termes qualitatiu i no pas quantitatiu.

8.2.Cohesió, pàgina 127

8.3.Acoblament, pàgina 128

Aproximacions quantitatives. Sovint les anàlisis les fem mesurant, de manera aproximada, la cohesió o l'acoblament en termes quantitatius. Però hem de ser sempre conscients que el que obtenim és una aproximació.

Anàlisi de l'acoblament. La *visibilitat indirecta* d'activació introdueix una nova dependència $\text{Caminada} \xrightarrow{atr} \text{Inscripció}$. La *visibilitat directa* d'activació introdueix una nova dependència $GLS \xrightarrow{atr} \text{Inscripció}$. El que al és analitzar aquestes dependències en termes d'acoblament.

Dependències i nous acoblaments. L'acoblament introduït per $\text{Caminada} \xrightarrow{atr} \text{Inscripció}$ es pot expressar amb $\text{Caminada} \rightarrow \text{Inscripció}$; i es tracta d'un acoblament no existent prèviament en el model. En canvi, l'acoblament introduït per $GLS \xrightarrow{atr} \text{Inscripció}$, que es pot expressar amb $\text{Caminada} \rightarrow \text{Inscripció}$, és un acoblament pre-existent prèviament en el model, gràcies a $GLS \xrightarrow{loc} \text{Inscripció}$.

Excursió. (Acoblament i dependències) Quan analitzem les dependències analitzem qui té coneixement de qui, en quin àmbit de visibilitat, i amb quina multiplicitat.

De cara a l'anàlisi de l'acoblament n'hi ha prou en analitzar qui coneix a qui. Així, una dependència significa que hi ha acoblament, però no necessàriament la introducció d'una nova dependència ha de significar que es produeix un acoblament nou. De fet, sovint un acoblament es manifesta amb més d'una dependència.

En el cas que ens ocupa, la visibilitat d'emmagatzematge exigida per l'emmagatzemador orb (*GLS*) introdueix el mateix acoblament que el que introdueix la visibilitat d'activació directa de la *i:Inscripció*. En ambdós casos tenim una dependència $GLS \rightarrow \text{Inscripció}$: multiavaluada d'atribut pel cas de l'emmagatzematge; i monoavaluada d'atribut pel cas de l'activació.

Per tant, l'activació amb *visibilitat directa* si bé introdueix la dependència
 $GLS \xrightarrow{atr} \text{Inscripció}$, no modifica l'acoblament, ja que prèviament ja teníem
la dependència $GLS \xrightarrow{loc} \text{Inscripció}$.

Anàlisi quantitativa de l'acoblament. Quantitativament doncs el model amb *visibilitat indirecta* d'activació està més acoblat: té un acoblament que l'altre model no té.

Anàlisi qualitativa de l'acoblament. Qualitativament sembla clar que exigir que la *c*:Caminada conegui la *i*:Inscripció significa que tenim més necessitats de coneixement i accés que en el cas que no ho exigim. Per tant, *qualitativament* el model amb visibilitat indirecta d'activació està més acoblat.²

Anàlisi superficial de la cohesió. Ambdues propostes (activació amb visibilitat directa o amb visibilitat indirecta) han de resoldre la mateixa responsabilitat: l'activació de la *i*:Inscripció. Per tant, en un anàlisi superficial podem concloure que la cohesió en ambdós casos és la mateixa.

Anàlisi afinada de la cohesió. En una anàlisi més afinada de la cohesió veiem que ambdues propostes exerceixen la responsabilitat inicial de maneres diferents. En el model amb visibilitat d'activació directa el controlador manté la *i*:Inscripció activa; en el model amb visibilitat d'activació indirecta el controlador manté una *c*:Caminada, i aquesta manté la *i*:Inscripció activa.

Anàlisi quantitativa de la cohesió. En el model amb visibilitat directa d'activació hi ha una responsabilitat a resoldre: el manteniment de $GLS \xrightarrow{atr} \text{Inscripció}$. En canvi en el model amb visibilitat indirecta d'activació hi ha dues responsabilitats a resoldre: el manteniment de $GLS \xrightarrow{atr} \text{Caminada}$ i el manteniment de $\text{Caminada} \xrightarrow{atr} \text{Inscripció}$. Per tant, en una anàlisi *quantitativa* el model amb visibilitat indirecta d'activació està més acoblat.

Anàlisi qualitativa de la cohesió. Des d'un punt de vista qualitatiu podem defensar que el manteniment tant de $\text{Caminada} \xrightarrow{atr} \text{Inscripció}$ com de $GLS \xrightarrow{atr} \text{Caminada}$ formen part de la responsabilitat de mantenir la *i*:Inscripció activa. Per tant, encara que comptabilitzem dues visibilitats a mantenir, estem resolent una única responsabilitat. En conseqüència podem defensar que ambdues propostes (activació amb visibilitat directa i activació amb visibilitat indirecta) tenen exactament la mateixa cohesió.³

²L'anàlisi qualitativa de l'acoblament pot discrepar de l'anàlisi quantitativa quan, per exemple, una informació semànticament atípicament l'expressem en el model amb més d'un component.

³En l'anàlisi superficial de la cohesió simplement hem dit que en ambdues propostes cal mantenir la *i*:Inscripció activa. En una anàlisi més afinada hem vist que en el cas de la visibilitat indirecta d'activació hi ha dues responsabilitats a resoldre, ja que hi ha dues visibilitats a mantenir. El que diem aquí, en l'anàlisi qualitativa de la cohesió, és que semànticament aquestes dues responsabilitats estan estretament acoblades; de fet, semànticament es tracta d'una única responsabilitat que en termes de disseny disgreguem en dues responsabilitats diferents. És aquest estret

Resultat de l'anàlisi de les propostes a $M1$. En conclusió, pel principi del *Baix Acoblament* i pel principi de l'*Alta Cohesió*, en el model $M1$ és preferible l'activació per *visibilitat directa*, ja que no introdueix nous acoblaments i la cohesió és major (en una anàlisi quantitativa) o idèntica (en una anàlisi qualitativa).

A $M1$ és preferible l'activació directa de la $i : \text{Inscripció}$

(36)

Anàlisi interrompuda. El fet d'usar la visibilitat indirecta $\text{GLS} \xrightarrow{\text{loc}} \text{Inscripció}$ té d'altres conseqüències que analitzarem més endavant. \rightarrow

13.4.1. Una solució poc eficient, pàgina 242

3 Activació a $M2$

3.1 Anàlisi prèvia de $M2$

Recordatori del model considerat $M2$. $\text{gls} : \text{GLS}$ és el controlador, $i c : \text{Caminada}$ té el rols de creadora i emmagatzemadora de la $i : \text{Inscripció}$.

Visibilitats actuals del model $M2$. En el model que estem considerant tenim les següents visibilitats:

$\text{GLS} \xrightarrow{\text{par}} \text{Caminada}$	Argument ES
$\text{Caminada} \xrightarrow{\text{loc}} \text{Inscripció}$	Creació
$\text{Caminada} \xrightarrow{\text{atr}} \text{Inscripció}^*$	Emmagatzematge

Visibilitats actuals sobre *Inscripció*. Qui té visibilitat sobre la $i : \text{Inscripció}$ és la $c : \text{Caminada}$. Però per a l'activació el que ens cal és una visibilitat monoavaluada d'atribut des del controlador.

Dues possibilitats. Per tant per mantenir la $i : \text{Inscripció}$ activa tenim dues possibilitats:^{4,5}

- *Visibilitat directa* $\text{GLS} \xrightarrow{\text{atr}} \text{Inscripció}$
- *Visibilitat indirecta* $\text{GLS} \xrightarrow{\text{atr}} \text{Caminada} \xrightarrow{\text{atr}} \text{Inscripció}$

3.2 Activació per visibilitat directa a $M2$

Visibilitat directa d'activació de la $i : \text{Inscripció}$. Com a visibilitat d'activació usem $\text{GLS} \xrightarrow{\text{atr}} \text{Inscripció}$.

acoblament entre les responsabilitats de mantenir aquestes dues visibilitats que ens permet afirmar que necessàriament han de ser considerades com una sola responsabilitat.

⁴La justificació d'aquestes possibilitats es realitza tot seguit dins l'anàlisi de cadascuna d'elles.

⁵Curiosament les possibilitats a $M1$ i a $M2$ són les mateixes.

Manteniment de l'activació directa a $M2$. El controlador delega en la $c: \text{Caminada}$ la creació de la $i: \text{Inscripció}$. Un cop creada la $i: \text{Inscripció}$ la $c: \text{Caminada}$ la comunica al controlador per tal que l'emmagatzemi de manera monoavaluada.

Visibilitats de $M2$ amb activació amb visibilitat directa. En el model que estem considerant tenim les següents visibilitats:

$\text{GLS} \xrightarrow{\text{par}} \text{Caminada}$	Argument ES
$\text{Caminada} \xrightarrow{\text{loc}} \text{Inscripció}$	Creació
$\text{Caminada} \xrightarrow{\text{atr}} \text{Inscripció}^*$	Emmagatzematge
$\text{GLS} \xrightarrow{\text{atr}} \text{Inscripció}$	Activació
$\text{Caminada} \xrightarrow{\text{par}} \text{Inscripció}$	Creació i manteniment de l'activació

Una dependència transformada. La dependència local resultat de la creació de la $i: \text{Inscripció}$ es transforma en una dependència de paràmetre per tal de poder comunicar enrere la $i: \text{Inscripció}$ acabada de crear. Per això hem suprimit la dependència local i n'hem fet aparèixer una de paràmetre.

Violació d'*Espill*. La visibilitat d'activació proposada no és el model de cap associació del MC. Per tant estem violant el principi de l'*Espill*. I això ho volem evitar de totes totes.

3.3 Activació per visibilitat indirecta a $M2$

Visibilitat d'activació de la $i: \text{Inscripció}$. Com a visibilitat d'activació usem la visibilitat indirecta $\text{GLS} \xrightarrow{\text{atr}} \text{Caminada} \rightarrow \text{Inscripció}$. Tot seguit analitzem com hem arribat a aquesta visibilitat, i quines implicacions té.

Introducció de la *Caminada* activa. Volem que el controlador tingui una visibilitat (in)directa (monoavaluada) sobre la $i: \text{Inscripció}$. Si convertim la visibilitat local que la *Caminada* té sobre la *Inscripció* acabada de crear en visibilitat d'atribut, per aconseguir l'objectiu n'hi ha prou en què el controlador tingui visibilitat directa sobre la $c: \text{Caminada}$. El resultat és una cadena de visibilitats que del controlador arriba a $i: \text{Inscripció}$ passant per $c: \text{Caminada}$. En d'altres paraules, *per aconseguir una Inscripció activa cal mantenir una Caminada activa*.

Activació de les *Inscripcions* local a cada *Caminada*. La visibilitat directa monoavaluada de *Caminada* sobre *Inscripció* es pot veure com l'activació d'una *Inscripció* dins de cada *Caminada*. La *Inscripció* activa per l'ES ser? la *Inscripció* activa dins la *Caminada* que l'ES veu com a activa.

Visibilitats de $M2$ amb activació amb visibilitat indirecta.

$GLS \xrightarrow{par} Caminada$	Argument ES
$Caminada \xrightarrow{lac} Inscripció$	Creació
$Caminada \xrightarrow{atr} Inscripció^*$	Emmagatzematge
$GLS \xrightarrow{atr} Caminada$	Activació
$Caminada \xrightarrow{atr} Inscripció$	Creació i activació

Una dependència transformada. La dependència local resultat de la creació de la $i:Inscripció$ es transforma en una dependència d'atribut per tal de mantenir emmagatzemada la $i:Inscripció$ acabada de crear. Per això hem suprimit la dependència local i n'hem fet aparèixer una d'atribut.

3.4 Comparativa de les dues possibilitats sobre $M2$

Anàlisi quantitativa i anàlisi qualitativa. Per comparar ambdues propostes (activació amb visibilitat directa o indirecta) cal analitzar els models resultants en termes de cohesió i acoblament. Tal i com hem vist més amunt, \rightarrow l'anàlisi ha de ser qualitativa, tot i que sovint en fem aproximacions quantitatives.

13.2.4.Comparativa de les dues possibilitats sobre $M1$, pàgina 237

Anàlisi de l'acoblament. La visibilitat directa d'activació introdueix un nou acoblament $GLS \xrightarrow{atr} Inscripció$, que a més viola *Espill*; en canvi la visibilitat indirecta $GLS \xrightarrow{atr} Caminada \xrightarrow{atr} Inscripció$ introdueix noves dependències però no introdueix nous acoblaments.^{6,7,→}

2.4.Excursió. (Acoblament i dependències), pàgina 237

Anàlisi de la cohesió. En termes de cohesió la millor solució és la visibilitat directa: en una anàlisi quantitativa té una responsabilitat menys; i en una anàlisi qualitativa hi ha motius per defensar que ambdues propostes tenen exactament les mateixes responsabilitats.^{8,→}

13.2.4.Comparativa de les dues possibilitats sobre $M1$, pàgina 237

Resultat de l'anàlisi de les propostes a $M2$. La violació d'*Espill* la considerem greu. Aquest ja seria motiu suficient per decantar-nos per l'activació per visibilitat indirecta. Però a més l'activació indirecta és preferible per acoblament. El preu a pagar és (potser)⁹ una lleugera pèrdua de cohesió.

A $M2$ és preferible l'activació indirecta de la $i:Inscripció$

(37)

⁶En aquest exemple l'anàlisi quantitativa i l'anàlisi qualitativa de l'acoblament coincideixen: les noves dependències (anàlisi quantitativa) són un canvi important en les necessitats de coneixement (anàlisi qualitativa).

⁷Per una discussió de quan una dependència nova és un nou acoblament o no, vegeu 2.4.Excursió. (Acoblament i dependències), pàgina 237.

⁸L'anàlisi acurada és la mateixa que la vista a 13.2.4.Comparativa de les dues possibilitats sobre $M1$, pàgina 237.

⁹El *potser* fa referència a si en l'anàlisi qualitativa considerem que mantenir les dues visibilitats és una mateixa responsabilitat o no.

4 Anàlisi de l'eficiència

4.1 Una solució poc eficient

Problemes d'eficiència. La solució amb activació per visibilitat indirecta és doblement ineficient, tant a *M1* com a *M2*. És ineficient en espai perquè exigeix tenir tantes *Inscripcions* actives com *Caminades* hi hagi al sistema; és ineficient en temps perquè totes les operacions sobre la *Inscripció* activa han de passar per *Caminada*.

Ineficiència en espai. Hem vist que la *c*:*Caminada* necessita visibilitat d'atribut sobre *i*:*Inscripció*; però els atributs es defineixen en termes de components, i no pas de realitzacions d'aquests. D'aquí que *tota* *Caminada* tingui visibilitat d'atribut (directa) sobre *Inscripció*. És a dir, tenim (potencialment) una *i*:*Inscripció* activa per cada *c*:*Caminada*.

Situació hipotètica. Potser en alguna extensió raonable dels requeriments del sistema això pot tenir sentit, però amb l'especificació que estem treballant el fet de mantenir una *Inscripció* activa és per assegurar que els ES del mateix CU saben amb quina *Inscripció* han de treballar. És a dir, l'àmbit d'"activació" d'una *Inscripció* és el CU (podem observar com el CU acaba sense deixar cap *Inscripció* activa). Així el que realment ens interessa és *una* *Inscripció* activa en cada moment (una per cada realització del CU). Per tant tenir atributs per mantenir una *Inscripció* activa per cada *Caminada* sembla un clar malbaratament d'espai.

Ineficiència en temps. El manteniment d'una *Inscripció* activa és per tal que els propers ES sàpiguen on han d'actuar. Però com que el controlador té visibilitat indirecta sobre aquesta *Inscripció* llavors tota petició feta per l'usuari hauria de ser transmesa prèviament a la *Caminada* activa, que és qui té la visibilitat sobre la *Inscripció* activa. El resultat és que s'allarga (potser innecessàriament) la cadena de missatges per respondre els ES.

Volem ser eficients. Si ens preocupem per l'eficiència llavors a *M2* tenim un problema: la solució proposada és la de l'activació per visibilitat indirecta; però aquesta solució és ineficient, i nosaltres volem ser eficients!

Eficiència en segon pla. El principi de la *Eficiència en segon pla* ens diu que en el moment de dissenyar ens hem d'oblidar de l'eficiència. Per tant les conclusions obtingudes per *M2* es mantenen vàlides, independentment de la ineficiència del resultat.¹⁰

4.2 Disseny sense anàlisi exhaustiva

El nostre camí. En aquest capítol hem analitzat per cada model, *M1* i *M2*, les dues propostes possibles: activació de la *i*:*Inscripció* amb visibilitat directa i amb visibilitat indirecta. En general per? el procediment no és aquest.

¹⁰Com veurem més endavant, l'eficiència sempre es pot aconseguir a posteriori. En el nostre cas, per exemple, els embolcalls permeten mantenir el model proposat i alhora introduir eficiència en espai i en temps.

El camí habitual. Habitualment el disseny és incremental. Per assignar una responsabilitat analitzem les diferents alternatives, i ens quedem amb el disseny que ens sembla millor. Per assignar la següent responsabilitat partim del model obtingut fins el moment, i només tornem enrere en cas que canviïn les condicions de context i calgui fer una reconsideració.

El camí habitual en el nostre cas. En l'exemple que ens ocupa, després d'aplicar *Controlador* i d'assignar la primera responsabilitat (la d'obtenir una nova *i:Inscripció* per a la *c:Caminada* donada) hem decidit que el model *M2* és preferible al model *M1*. Ara sobre aquest model *M2* cal assignar la responsabilitat de mantenir la *i:Inscripció* activa (mentre no hi hagi *reconsideració* el model *M1* el considerarem bandejat).

Una assignació problemàtica. Partim doncs de *M2*. L'anàlisi[→] de l'assignació de la nova responsabilitat ens porta a decantar-nos per l'activació de la *i:Inscripció* mitjançant una visibilitat d'activació indirecta que passa per la *c:Caminada*. Però la decisió no ens agrada perquè la veiem ineficient.

13.3.Activació a *M2*, pàgina 239

Asseveració falsa. Dir que el model resultant no ens convenç perquè no és eficient demostra que ens hem oblidat del principi de l'*Eficiència en segon pla*.

Una hipòtesi de treball. Anem però a analitzar què passaria si realment ens oblidem del principi de l'*Eficiència en segon pla* i ens entestem en obtenir un disseny eficient.

4.3 Anem a violar l'*Eficiència en segon pla*

Podem modificar el disseny obtingut de *M2* de tal manera que no tinguem els problemes d'ineficiència detectats?

(29) ?

Millorar l'eficiència en temps. Per millorar l'eficiència en temps caldria convertir la visibilitat indirecta d'atribut de la *Inscripció* per part del controlador, en una *visibilitat directa*. Per fer-ho cal que la *Caminada* retorni al controlador la *i:Inscripció* que ha creat. Però llavors el controlador queda acoblat a la *i:Inscripció*.

De l'eficiència en temps al desastre. Per assignar la responsabilitat de crear la *i:Inscripció*¹¹ hem analitzat els seus emmagatzemadors canònics.[→] Si afegim ara que el controlador ha de mantenir una visibilitat d'atribut sobre la *i:Inscripció*, visibilitat que viola el principi de l'*Espill*, llavors l'anàlisi de l'acoblament generat pels diferents creadors variar? sensiblement. En concret l'acoblament exigint en l'ús de creadors no canònics pot ser un acoblament ja present, i per tant admissible. El resultat és que el quadre de decisions de

11.1.3.Creador canònic, pàgina 202

¹¹Pregunta 13., pàgina 121.

Figura 11.1, pàgina 213

la figura 11.1, pàgina 213, \rightarrow deixa de ser correcta. Tot seguit ho veiem amb deteniment.

Canvis en el context de l'anàlisi. En el moment que proposem la visibilitat d'activació $GLS \xrightarrow{atr} Inscripció$ estan canviant les condicions d'anàlisi que ens han portat a decantar-nos per la *c*:Caminada com a *creadora canònica* de la *i*:Inscripció, i que hem expressar en la hipòtesi 24 de la pàgina 214. \rightarrow

Hipòtesi 24, pàgina 214

Visibilitats actuals sobre Inscripció. En el model *M2* que estem considerant qui té visibilitat sobre la *i*:Inscripció és la *c*:Caminada. Per una banda tenim visibilitat local, com a mínim, de la *i*:Inscripció acabada de crear; per l'altra visibilitat per multiobjecte (o visibilitat multiavaluada) de totes les "seves" Inscripcions.

Creador orb i creador canònic. En la presentació dels principis de creació \rightarrow dèiem que era preferible emprar un creador canònic ja que la visibilitat d'emmagatzematge coincidia amb la visibilitat que modelitza una de les associacions del MC. I que l'ús d'un creador orb (no canònic) introduïa una visibilitat que violava *Espill*.

10.9.Ús dels diferents principis de creació, pàgina 187

Canvis en les condicions d'anàlisi. En el cas que ens ocupa, la visibilitat d'emmagatzematge exigida per l'emmagatzemador orb (*GLS*) introdueix el mateix acoblament que el que introdueix la visibilitat d'activació directa de la *i*:Inscripció. En ambdós casos tenim una dependència $GLS \rightarrow Inscripció$: multiavaluada d'atribut pel cas de l'emmagatzematge; i monoavaluada d'atribut pel cas de l'activació. És a dir, l'acoblament introduït pel *creador orb* és el mateix que introdueix l'activació per *visibilitat directa* de la *i*:Inscripció.

Reconsideració. El canvi de les condicions d'anàlisi (en concret, la presència de $GLS \xrightarrow{atr} Inscripció$) obliga a reconsiderar el creador d'*i*:Inscripció. I en aquest cas observem com el creador orb (*GLS*) no introdueix nous acoblaments; mentre que el creador canònic (*c*:Caminada) introdueix un nou acoblament: el de la visibilitat d'emmagatzematge *Caminada* $\xrightarrow{atr} Inscripció^*$.

Resultat de la consideració. Reconsiderem el creador d'*i*:Inscripció, i assignem el rol de la creació a *GLS*. Però llavors estem retornant al model *M1*!!

Conseqüències de l'intent de millora de l'eficiència en temps. Per millorar l'eficiència en temps hem decidit emprar la visibilitat directa per a l'activació de la *i*:Inscripció. Però això obliga a reconsiderar l'aplicació dels principis de creació. En aquest cas la reconsideració ens diu que l'ús del controlador com a creador dóna un disseny amb un acoblament més baix.

Compte no caiguem a la trampa. Cal anar molt en compte de no caure en el parany. Si estem entossudits en l'eficiència, però no ens adonem que els canvis introduïts per mor de l'eficiència canvien el context de les decisions anteriors, i per tant no apliquem la reconsideració allà on calgui, llavors ens trobem amb

un disseny erroni ja que *les argumentacions considerades per prendre les decisions de disseny anteriors es converteixen en falses argumentacions*.

Millora de l'eficiència en espai. La solució considerada per millorar l'eficiència en temps també millora l'eficiència en espai: només mantenim una *Inscripció activa*, que és la que el controlador en té visibilitat d'atribut directa. Però llavors, com abans, les argumentacions que sustenten el disseny deixen de ser vàlides.

L'eficiència prematura ens pot fer canviar el context de les decisions prèvies, i per tant, una de dues:

- O hem de reconsiderar tot el disseny
- O el disseny proposat es fonamenta en argumentacions falses

(38)

4.4 Què fem davant la manca d'eficiència

Davant un cas com el present, on les decisions perfectament argumentades segons els principis de disseny ens porten a un disseny ineficient, no ens queda més que apuntar-nos la manca d'eficiència i els seus motius.

Teranyina. El principi de la *Teranyina*[→] ens diu que hem de veure el disseny en tota la seva globalitat: intentar a hores d'ara modificar i forçar el disseny per fer-lo eficient pot ser que ens porti a una bona eficiència local, però potser els canvis introduïts localment poden fer que globalment el sistema deixi de ser eficient. D'aquí que el principi de l'*Eficiència en segon pla* ens diu que primer ens preocupem de dissenyar *tot* el sistema (o subsistema), i que després ja ens preocuparem de donar eficiència allà on calgui.

2.4.3. Una densa i fràgil teranyina de decisions, pàgina 27

Documentació de l'eficiència. En l'exercici que estem desenvolupant caldrà introduir la següent anotació:

En el model *M2* on *Caminada* és creadora canònica (i per tant, també emmagatzemadora) d'*Inscripció* tenim:

(30) !

Ineficiència en el temps. Les peticions sobre *Inscripció* han de passar per *Caminada*

Ineficiència en espai. Tenim una *Inscripció activa* per cada *Caminada*

5 Conclusions i decisions de disseny

Resum de l'anàlisi. En el cas d'usar el controlador com a creador i emmagatzemador de les *Inscripcions* (principi *Creador orb*), el mateix controlador

pot mantenir la *Inscripció* activa sense problemes.¹² La visibilitat directa sobre la *Inscripció* activa ens dóna eficiència tant en temps com en espai. Per tant, en termes d'eficiència (local) aquest model és superior al disseny on *Caminada* és creadora i emmagatzemadora d'*Inscripció*. Però com hem vist, amb el coneixement actual del sistema, els principis de disseny es decanten pel model "no eficient".

! (31)

Eficiència del repositori centralitzat El model amb repositori centralitzat d'*Inscripcions* visible des del controlador és eficient en espai i temps. Però viola els principis d'*Encarrilament*, *Alta cohesió*, *Baix acoblament*, *Es-pill* i *Creador can?nic*.

Ineficiència del repositori distribuït El model amb repositori distribuït d'*Inscripcions* en les diferents *Caminades* és ineficient en temps i espai \rightarrow . Per? té el suport dels principis de disseny.

Observació 30, pàgina 245

Conclusió i decisió de disseny. De tot plegat podem prendre la següent decisió, que marquem com *alternativa* ja és un possible punt de reconsideració. Per prendre aquesta decisió continuem mantenint com a vàlides les hipòtesis de 24, pàgina 214 \rightarrow

Hipòtesi 24, pàgina 214

🔗 (32)

La *i*:*Inscripció* activa la mantenim amb visibilitat indirecta:

El controlador ha de mantenir una *Caminada* activa. Cada *Caminada* ha de mantenir una *Inscripció* activa.

Intermediari de la visibilitat. La *c*:*Caminada* és qui fa d'intermediari en la visibilitat necessària per mantenir la *i*:*Inscripció* activa.

Canvis en els contractes. Seguint el principi de l'*Estabilitat de l'especificació* \rightarrow els contractes no es modifiquen, i simplement afegim una nota sobre la decisió que hem pres: activació indirecta passant per *Caminada*.

12.2.Manteniment consistent dels contractes, pàgina 224

¹²Ens referim a què la introducció de la *Inscripció* activa no viola cap dels principis de disseny; i en concret, que l'*acoblament* i la *cohesió* no es veuen alterats.

6 Els models en la situació actual

Creació de la *i*: **Inscripció**. Repetim aquí la figura 11.1, pàgina 213: →

Figura 11.1, pàgina 213

Responsable	Principi	Observacions
GLS	<i>Expert</i> <i>Creador orb</i> Viola: <i>Encarrilament</i> <i>Creador canònic</i>	<ul style="list-style-type: none"> • Repositori centralitzat d'Inscripcions • “referent <i>a</i>”: $\text{Inscripció}^{\text{atr}} \rightarrow \text{Caminada}$ • Necessitat de recuperació de la Caminada • Recuperació: Cal passar la Caminada al mètode de creació <p>Cal una visibilitat $\text{GLS}^{\text{atr}} \rightarrow \text{Inscripció}^*$, que no es correspon a cap associació de MC</p>
c: Caminada	<i>Expert</i> <i>Baix acoblament</i> <i>Creador canònic</i>	<ul style="list-style-type: none"> • Repositori d'Inscripcions fragmentat per Caminades • “referent <i>a</i>”: $\text{Caminada}^{\text{atr}} \rightarrow \text{Inscripció}^*$

Figura 13.1: Creadors d'*i*: Inscripció segons *Expert* i *Creador*

Activació de la i:Inscripció. La següent figura mostra les decisions sobre l'activació de la i:Inscripció:

Model M1		
Responsabilitat	Responsable	Justificació
Creació i:Inscripció	GLS	Expert Creador Orb <i>Encarrilament</i> <i>Creador canònic</i>
Activació i:Inscripció	GLS	Activació directa GLS→Inscripció

Model M2		
Responsabilitat	Responsable	Justificació
Creació i:Inscripció	Caminada	Expert Creador canònic
Activació i:Inscripció	GLS, Caminada	Activació indirecta GLS→Caminada→Inscripció

Figura 13.2: Descripció dels models amb l'activació de la i:Inscripció

Part III

Terres pantanoses

Part III: Taula de continguts

14 Encontres en la tercera fase	253
1 Esbós del camí	257
2 Un problema comunicatiu	257
3 Mons comunicables	259
4 Un llenguatge comú	263
5 Comunicació i principis de modelització	267
6 La comunicació és possible	275
Definicions	276
15 Identificadors	277
1 Esbós del camí	282
2 Llenguatge per nivells	282
3 Propietat d'identificació	288
4 Component ben definit	296
5 Identificació i llenguatge de comunicació	298
6 La comunicació és possible	313
Definicions	315

Capítol 14

Encontres en la tercera fase

1	Esbós del camí	257
2	Un problema comunicatiu	257
2.1	El problema de partida	257
2.2	Generalització del problema comunicatiu	258
3	Mons comunicables	259
3.1	Completesa i solidesa	259
3.2	Coherència	261
4	Un llenguatge comú	263
4.1	Llenguatge comú de comunicació	263
4.2	Un llenguatge de <i>noms</i>	264
4.3	Propietats del llenguatge comú	265
5	Comunicació i principis de modelització	267
5.1	Objectiu que perseguim	267
5.2	Models comunicables	268
5.3	Llenguatge de noms i modelització	269
5.4	Conclusions: principis de modelització i comunicació	274
6	La comunicació és possible	275
	Definicions	276

Contingut detallat del capítol 14

1	Esbós del camí	257
	Problema comunicatiu	257
	Models i realitats comunicables	257
	Llenguatge de comunicació	257
	El paper dels principis de desenvolupament	257
2	Un problema comunicatiu	257
2.1	El problema de partida	257
	Col·lisió entre dos mons	257
	Dues llengües diferents	257
	Dues realitats diferents	257
	Estratègia per resoldre el problema comunicatiu	258
2.2	Generalització del problema comunicatiu	258
	Mons que volem comunicar	258
	El sistema com a model	258
	L'exterior del sistema	258
	Comunicació per resoldre	258
3	Mons comunicables	259
	Anàlisi de les propietats dels mons comunicables	259
3.1	Completesa i solidesa	259
3.1.1	La relació entre els dos mons	259
	Completesa i solidesa en termes planers	259
	Apunt sobre la notació	259
3.1.2	Comunicació imperfecta	260
	Ponts per a la comunicació	260
	Passos fronterers	260
	Existència i unicitat	260
	Conseqüències de la no unicitat de la correspondència	260
	Comunicació imperfecta	261
	Pèrdua d'informació	261
	Manca d'informació	261
	Comunicació possible, però imperfecta	261
3.2	Coherència	261
	Isomorfia entre els dos mons	261
	Completesa i isomorfia	262
	Solidesa i isomorfia	262
	Comunicació perfecta	262
	Condició necessària	263
	Estructura d'un món	263
	Isomorfia i comunicació	263
4	Un llenguatge comú	263
4.1	Llenguatge comú de comunicació	263
	Cal resoldre la comunicació	263
	Necessitat d'un llenguatge comú	263

	Una sintaxi amb dues semàntiques	264
	Necessitat d'una moneda de canvi	264
4.2	Un llenguatge de <i>noms</i>	264
	Universalitat dels noms	264
	Llenguatge de noms	264
	Noms del model i de la realitat	264
	Extensió dels noms	264
	Noms del model i noms de la realitat	265
4.3	Propietats del llenguatge comú	265
4.3.1	Vides paral·leles	265
	Un nom, dues lectures	265
	Consistència denominadora	265
4.3.2	Ambigüïtat denominadora	265
	No ambigüïtat	265
	Ambigüïtat denominadora en la realitat	265
	Ambigüïtat denominadora en el model	266
	Consistència i ambigüïtat	266
4.3.3	Anòmia	266
	Elements sense nom	266
4.3.4	Propietats del llenguatge comú de comunicació	267
4.3.5	Condicions necessàries per a la comunicació	267
	Estructura i llenguatge	267
	Condicions suficients per a la comunicació	267
5	Comunicació i principis de modelització	267
5.1	Objectiu que perseguim	267
	El paper dels principis en la comunicació	267
	Suficiència dels principis de modelització	267
5.2	Models comunicables	268
	Completesa i solidesa	268
	Imperfecció en la comunicació	268
	Perfecció en la comunicació	268
	Coherència	268
	El principi <i>Espill</i>	268
	El que li manca a <i>Espill</i>	268
	<i>Completesa</i>	268
5.3	Llenguatge de noms i modelització	269
5.3.1	Noms d'un element	269
	Multiplicitat de noms	269
	Canvis exigits per la multiplicitat dels noms	269
5.3.2	Semàntiques consistents	269
	Semàntiques consistents en termes planers	269
	<i>Franquícia</i>	269
	<i>Franquícia obligada</i>	269
	El significat de "poder usar"	269
	<i>Franquícia obligada</i> i sinonímia	270
5.3.3	Sintaxi comuna	270

	Conjunt de noms	270
	Suficiència dels noms compartits	270
	Sintxi comuna	272
	Sintaxi comuna per a la comunicació	272
5.3.4	Referent únic	272
	Recordatori: La propietat del <i>Referent únic</i>	272
	Recordatori: El principi del <i>Referent únic</i>	272
	Propietat i principi	272
	Referent únic en la realitat	272
	El paper del glossari	272
	Referent únic en el model	272
	Necessitat del principi del <i>Referent únic</i>	273
5.3.5	Completesa del conjunt de noms	273
	Recordatori: la <i>Franquícia obligada</i>	273
	La <i>Franquícia obligada</i> prohibeix les anòmies en la realitat	273
	La <i>Franquícia obligada</i> prohibeix les anòmies en el model	273
	Completesa del conjunt de noms	274
5.4	Conclusions: principis de modelització i comunicació	274
5.4.1	Model comunicable	274
	La responsabilitat del desenvolupador	274
5.4.2	Conjunt de noms vàlid per a la comunicació	274
	Noms per a la comunicació	274
	Principis necessaris per a la comunicació	274
6	La comunicació és possible	275
	Conclusions planeres	275
	Comunicació per noms	275
	Supressió dels elements sense nom	275
	Sabem què vol dir cada nom	275
	Què comuniquem	275
	Selecció dels noms	275
	Definicions	276

1 Esbós del camí

Aquest capítol és fonamentalment tècnic. I per tant complex. En una primera lectura es pot passar per alt, a excepció del present apartat, on s'explica què es pretén i perquè; i del darrer, on s'exposen les conclusions en termes planers.

(33) !

Problema comunicatiu. Tenim un problema comunicatiu: la comunicació de dos mons; la comunicació entre model i realitat; la comunicació entre problema i solució. (Apartat 14.2. [Un problema comunicatiu](#), pàgina 257).

Models i realitats comunicables. Ara bé, només té sentit comunicar allò que en l'altre món pot tenir algun significat. És a dir, no tot model ni tota realitat són aptes per a la comunicació. (Apartat 14.3. [Mons comunicables](#), pàgina 259).

Llenguatge de comunicació. Un cop sabem que l'estructura d'ambdós mons permet la comunicació, cal definir un llenguatge comú que permeti realitzar la comunicació. Aquest llenguatge comú és un conjunt de noms que ha de tenir unes propietats determinades. (Apartat 14.4. [Un llenguatge comú](#), pàgina 263).

El paper dels principis de desenvolupament. Els principis de desenvolupament, si es segueixen al peu de la lletra, asseguren que l'estructura del model obtingut és l'exigida per a la comunicació, i que el conjunt de noms del model que també apareixen en la realitat tinguin les propietats necessàries per esdevenir el llenguatge de comunicació cercat. (Apartat 14.5. [Comunicació i principis de modelització](#), pàgina 267).

2 Un problema comunicatiu

2.1 El problema de partida

Col·lisió entre dos mons. Els ES comuniquen dos móns irreconciliables: el món exterior (on viu el problema) i el món del sistema (on viu la solució software).^{1, →}

2.1.2. Lectura dels contractes dels ES, pàgina 22

Dues llengües diferents. Les peticions fetes pels ES han de ser fetes en termes intel·ligibles per l'agent extern que genera l'ES; el disseny dels ES ha de resoldre la petició manipulant elements de la solució software proposada. És a dir, qui fa la petició i qui la respon són agents de mons diferents, i parlen llengües diferents.

¹A 2.1.2. [Lectura dels contractes dels ES](#), pàgina 22, hem plantejat que els ES tenen dues lectures, la que correspon a l'especificació i la que correspon al disseny. Aquí aprofundim en aquestes dues lectures, i en la seva coherència: no s'hi val, per exemple, que la lectura feta pel dissenyador distorsioni totalment els propòsits de l'especificador que ha escrit el contracte.

Dues realitats diferents. Però no només és un problema de llenguatge; també hi ha un problema de realitats diferents: els elements que manipula el disseny són realitzacions dels components; però aquestes no poden viure fora del sistema. Una simple traducció de llengües, paraula per paraula, no serveix per mor de les realitats diferents que manipula cada una de les llengües en consideració.

- ? (34) Com resollem la necessitat de comunicar dos mons (el sistema i el món extern) que usen "realitats" diferents?

Estratègia per resoldre el problema comunicatiu. Per tal de resoldre aquesta pregunta cal analitzar quins són exactament els mons que cal comunicar; quins elements d'aquests mons volem comunicar; i quin mecanisme de comunicació s'ha d'emprar.

2.2 Generalització del problema comunicatiu

- ? (35) Quins són exactament els mons que volem comunicar?

Mons que volem comunicar. Els mons que volem comunicar són el món *exterior* al sistema i el món *interior* del sistema: $Exterior \leftrightarrow Sistema$.

El sistema com a model. El sistema és la solució d'un determinat problema; i aquesta solució l'hem construïda com un model d'aquest problema. El problema és per tant la realitat que el sistema modelitza.²

$$model(Problema) = Solució = Sistema$$

L'exterior del sistema. Els agents externs al sistema, els actors, interactuen amb el sistema a través dels ES en termes intel·ligibles per ells; és a dir, en termes del problema. Per tant la comunicació $Exterior \leftrightarrow Sistema$ és de fet una comunicació $Problema \leftrightarrow Sistema$.

Comunicació per resoldre. Per tant, la comunicació que volem establir entre el sistema i el món exterior a ell ($Exterior \leftrightarrow sistema$) és de fet una comunicació entre la realitat (el problema) i el model (la solució):

²De fet el sistema és el model del problema donat en l'especificació; però aquesta és un model de la realitat. Per tant el sistema és el model d'un model de la realitat.

Problema comunicatiu que cal resoldre

$$\begin{aligned} \text{Exterior} &\leftrightarrow \text{Sistema} \equiv \\ \text{Problema} &\leftrightarrow \text{Solució} \equiv \\ \text{Realitat} &\leftrightarrow \text{Model} \end{aligned}$$

(39)

3 Mons comunicables

Anàlisi de les propietats dels mons comunicables. El problema comunicatiu que cal resoldre és el problema $\text{Realitat} \leftrightarrow \text{Model}$. Començarem per analitzar quines propietats ha de tenir un model sobre una realitat per tal que la comunicació pugui ser possible. Més endavant[→] analitzarem el llenguatge de comunicació.

14.4. Un llenguatge comú, pàgina 263

3.1 Completesa i solidesa

3.1.1 La relació entre els dos mons

Definició. Completesa

Direm que un model MODEL és complert respecte una realitat REALITAT si i només si tot element de la realitat REALITAT és modelitzat amb algun element del model en qüestió:

$$\text{Completesa: } (\forall A \in \text{REALITAT}) (\exists B \in \text{MODEL}) B \in \text{model}(A)$$

Definició. Solidesa

Direm que un model MODEL és sòlid respecte una realitat REALITAT si i només si cadascun dels elements del model MODEL és el model d'algun dels elements de la realitat modelitzada:

$$\text{Solidesa: } (\forall B \in \text{MODEL}) (\exists A \in \text{REALITAT}) B \in \text{model}(A)$$

Completesa i solidesa en termes planers. La *completesa* diu que model “captura” tots els elements pertinents de la realitat. La *solidesa* diu que el model no “captura” cap element indesitjat.

Apunt sobre la notació. En les definicions de *completesa* i *solidesa* apareix $B \in \text{model}(A)$. En aquestes definicions, per tant, admetem que el model d'un element A de la realitat, $\text{model}(A)$, sigui un conjunt d'elements; d'aquí la petició que B pertanyi al model de A .^{3, →}

Exemple 64. Model sòlid i complert sense unicitat, pàgina 260

³A l'exemple 64. Model sòlid i complert sense unicitat, pàgina 260, es veu la necessitat d'expressar el model d'un element de la realitat en forma de conjunt.

3.1.2 Comunicació imperfecta

Ponts per a la comunicació. La solidesa i la completesa d'un model per una determinada realitat asseguruen que tot element del model té com a mínim un correlat en la realitat, i que tot element de la realitat té com a mínim un correlat en el model. És a dir, la solidesa i la completesa d'un model per aquesta realitat asseguruen l'existència de ponts que parteixen de tots i cadascun dels elements dels dos mons en consideració, i que porten a algun element de l'altre món.

Passos fronterers. Els ponts entre els diferents elements dels dos mons en consideració, que asseguruen les propietats de *completesa* i *solidesa* del model pel que fa a aquesta realitat, són el mecanisme que permet traspasar les fronteres d'ambós mons.

Existència i unicitat. Un model *sòlid* i *complert* assegura que tot element en un món té un correlat en l'altre. Però aquest correlat no necessàriament és únic: els ponts comunicatius poden tenir múltiples entrades i múltiples sortides.

Exemple 64 (Model sòlid i complert sense unicitat) *Suposem una realitat 3D formada per un cub i un con. El nostre model 2D l'obtenim per projecció. Per mor de la posició dels elements de la realitat, i per mor de la direcció de projecció obtenim que el con es projecta en una forma triangular; mentre que el cub es projecta sobre una forma quadrada que tapa la projecció del con: a la vista queda la punta superior (en forma de triangle), i part de la base (en forma de trapezi).*

Així el model consta de tres elements: un triangle, un quadrat, i un trapezi, tals que:

1. $model(cub) = \{quadrat\}$
2. $model(con) = \{quadrat, trapezi, triangle\}$

Cal observar com el con es modelitza amb tres elements i com el quadrat és el model de dos elements: no hi ha unicitat en la correspondència entre model i realitat. Tot i així tot element del model es correspon a algun element de la realitat; i tot element de la realitat es correspon amb algun element del model. Per tant el model és sòlid i complert per a aquesta realitat.⁴

Conseqüències de la no unicitat de la correspondència. La manca d'unicitat en la correspondència entre els elements del model i de la realitat modelitzada fan que, en usar aquesta correspondència amb finalitats comunicatives, les distincions que eren rellevants en un món poden deixar de ser-ho en l'altre. En termes metafòrics, el bagatge que portem en començar la travessa del pas fronterer pot ser que ens sigui espoliat pels agents duaners, o que ens sigui del tot insuficient per superar el fred de les gèlides temperatures de la serralada que cal atravesar.

⁴Certament hi ha moltes d'altres maneres de modelitzar aquesta projecció. Per exemple, enlloc de dividir-la en tres formes bàsiques la podríem haver dividit en cinc, subdividint la projecció del cub en tres: dos elements que no es superposen a la projecció del con, i un element que sí.

Exemple 65 (Manca d'informació) En el model de l'exemple 64 \rightarrow , distingim entre el *trapezi* i el *triangle*. Si volem comunicar qualsevol d'aquests elements a la realitat, només podrem arribar a comunicar la idea de *con*: tant si comuniquem *trapezi* com si comuniquem *triangle* en la realitat només podrem entendre *con*; el motiu és que en el model fem unes distincions que no són possibles en la realitat. En la comunicació perdem informació; els duaners ens han espoliat.

En canvi si volem comunicar l'element *quadrat* tant podem comunicar la idea de *con* com la de *cub*. La comunicació ens exigeix una diferenciació per la qual no estem preparats; no anem prou abrigats per passar la serralada.

Comunicació imperfecta. La manca d'unicitat en la correspondència entre el model i la realitat pertinent produeix "errors" en la comunicació: *pèrdues d'informació* o *mancances d'informació*.

Pèrdua d'informació. La *pèrdua d'informació* apareix allà on hi ha distincions que són pertinents en la realitat que no podran ser comunicades al model, o a la inversa: volem dir-te "a o b" allà on tu parles de "x".

Manca d'informació. El problema de la *manca d'informació* apareix allà on l'element que es vol comunicar té més d'un correlat en l'altre món: volem dir-te "a" allà on tu parles de "x o y".

Comunicació possible, però imperfecta. De tot plegat resulta que donat un model *sòlid i complert* per a una determinada realitat la comunicació és possible tot i que potser imperfecta. La possibilitat de la comunicació l'asseguren els ponts que fan de passos fronterers i posen en contacte els elements d'ambdós mons; la imperfecció en la comunicació és resultat de la manca d'unicitat en la correspondència entre els elements d'ambdós mons.

Mons comunicables

La comunicació entre realitat i model només és pertinent (tot i que segurament imperfecta) si el model és *complert i sòlid* per a aquesta realitat.

(40)

3.2 Coherència

Isomorfia entre els dos mons. Per evitar errors en la comunicació cal que cadascun dels elements comunicats només tingui un correlat en l'altre món: "volem dir-te 'a' allà on tu parles de 'x'". És a dir, la comunicació perfecta (sense errors) només és possible si la realitat i el seu model són *mons isomòrfs*: cada element de la realitat té un únic correlat en el model; i cada element del model té un únic correlat en la realitat.

Definició. Isomorfia entre model i realitat

Direm que el model i la realitat són isomorfs si i només si :

- $(\forall A \in \text{REALITAT}) [(\exists B \in \text{MODEL})(B \in \text{model}(A)) \rightarrow \text{model}(A) = B]$
- $(\forall B \in \text{MODEL}) [(\exists A \in \text{REALITAT})(B \in \text{model}(A)) \rightarrow \text{model}(A) = B]$

Completesa i isomorfia. La *completesa* diu que el MODEL “captura” tots els elements pertinents de la REALITAT. La *isomorfia* diu que quan el MODEL “captura” algun element de la REALITAT ho fa sense introduir “soroll”, és a dir, que cada element de la REALITAT es modelitza amb exactament un element del MODEL, i per tant aquest deixa de ser un conjunt:

Completesa i isomorfia:

$$(\forall A \in \text{REALITAT})(\exists B \in \text{MODEL}) \text{model}(A) = B$$

Solidesa i isomorfia. La *solidesa* diu que el MODEL no “captura” cap element indesitjat. La *isomorfia* diu que quan el MODEL “captura” un element de la REALITAT ho fa sense “barrejar” elements diferents d’aquesta, és a dir, que cada element del MODEL modelitza exactament un element de la REALITAT, i per tant el model deixa de ser un conjunt:

Solidesa i isomorfia:

$$(\forall B \in \text{MODEL})(\exists A \in \text{REALITAT}) B \in \text{model}(A) = B$$

Comunicació perfecta. Hem vist \rightarrow com la comunicació entre un model i una realitat només té sentit si el model és *sòlid* i *complet* respecte aquesta realitat. Si a més afegim la condició d’*isomorfia* entre els dos mons, obtenim les condicions necessàries per a la comunicació perfecta: la correspondència 1-1 entre model i realitat evita tant la pèrdua com la manca d’informació.

Definició. Model coherent

Direm que un model és coherent amb una determinada realitat si i només si :

- **Completesa unívoca**
 $(\forall A \in \text{REALITAT})(\exists B \in \text{MODEL}) \text{model}(A) = B$
- **Solidesa unívoca**
 $(\forall \text{MODEL})(\exists A \in \text{REALITAT}) \text{model}(A) = B$

Cohèrència: Propietats necessàries per a la comunicació perfecta

(41)

La comunicació perfecta entre realitat i model només és possible si el model és *coherent* respecte aquesta realitat. És a dir, si:

1. Cada element de la realitat té un i només un correlat en el model⁵
2. Cada element del model té un i només un correlat en el model⁶

Condició necessària. Cal observar com la *coherència* és una condició necessària per a la comunicació, tot i que no suficient. Com a mínim manca per definir les condicions que ha de complir el llenguatge de comunicació. →

14.4. Un llenguatge comú, pàgina 263

Estructura d'un món. L'estructura d'un món el està formada pels seus elements i per les interrelacions entre aquests. Però les interrelacions entre els diferents elements no deixen de ser elements del món considerat, i per tant l'estructura d'un món la podem veure com un conjunt d'elements. Vistes així les coses, la correspondència 1 – 1 dels elements preserva l'estructura, ja que els lligams estructurals es tracten com a elements qualssevol. Així, l'exigència d'*isomorfia* el que ens diu és que els dos mons a comunicar han de tenir la mateixa estructura.

Isomorfia i comunicació. És precisament la isomorfia entre model i realitat el que permet la comunicació perfecta: *quan en un món fem referència a un element, aquest es correspon amb un i només un element de l'altre món.*

4 Un llenguatge comú

4.1 Llenguatge comú de comunicació

Cal resoldre la comunicació. La pregunta 34, pàgina 258, → planteja la problemàtica de com comunicar dos mons amb realitats diferents. La conclusió 41, pàgina 263, → diu que aquesta problemàtica només es pot resoldre si els mons que volem comunicar tenen una mateixa estructura. Queda però resoldre la qüestió de com es realitza aquesta comunicació.

Pregunta 34, pàgina 258

Conclusió 41, pàgina 263

Quin mecanisme usem per fer possible la comunicació?

(36) ?

⁵De fet n'hi ha prou en què tinguin un i només un correlat en el model els elements de la realitat que volem comunicar.

⁶De fet n'hi ha prou en què tinguin un i només un correlat en la realitat els elements del model que volem comunicar. Si es compleix *Espill* sabem que cada element del model té un correlat en la realitat; i llavors només cal afegir que, per aquells elements que volguem comunicar, aquest correlat ha de ser únic. A 14.5. Comunicació i principis de modelització, pàgina 267 →, ho analitzem amb deteniment.

14.5. Comunicació i principis de modelització, pàgina 267

Necessitat d'un llenguatge comú. La idea bàsica per resoldre la comunicació entre dos mons diferents és la d'exigir l'existència d'un *llenguatge comú* en els dos mons.⁷

Una sintaxi amb dues semàntiques. El que es persegueix és la definició d'una sola sintaxi que es pugi interpretar tant en termes dels elements d'un dels mons, com en termes dels elements de l'altre món.

Necessitat d'una moneda de canvi. Per tal que la comunicació sigui possible, els elements comunicats, és a dir, aquells que traspassen les barreres dels mons, han de ser elements que puguin viure en ambdós mons. Com que les realitats subjacents a cadascun dels mons que volem comunicar són massa distintes, no podem usar els propis elements dels mons per a la comunicació; ens cal una espècie de moneda de canvi: els *noms*.

4.2 Un llenguatge de noms

Definició. Nom

*Un nom és una cadena fònica o textual. A tots els efectes un nom no té estructura interna ni comportament associat; un nom és una etiqueta arbitrària que podem aplicar a un element.*⁸

Universalitat dels noms. Els noms són valors *universals*, en el sentit que poden existir en qualsevol dels mons imaginables.

Llenguatge de noms. El llenguatge comú ha de ser un *llenguatge de noms* (pròpiament, ha de ser un conjunt de noms).

Noms del model i de la realitat. Direm que un nom és del model (o de la realitat) quan s'usi per expressar algun element del model (o, respectivament, de la realitat). Quan ens interressi remarcar que el nom és un nom del model o un nom de la realitat usarem $nom_M(x)$ i $nom_R(x)$ respectivament.

Extensió dels noms. El conjunt de tots els noms de les realitzacions d'un element A l'expressarem amb $NOMS_A$:

$$NOMS_A = \bigcup_{x \in A} nom(x)$$

⁷Aquesta exigència és independent de quina estructura tenen els mons. Hem vist però, que per tal que la comunicació sigui perfecta (és a dir, sense pèrdua ni manca d'informació), cal que els mons que volem comunicar siguin isomorfs.

⁸Més endavant[→] veurem que pròpiament els noms són una seqüència d'etiquetes.

Exemple 66 (Extensió dels noms) *Sigui Client un concepte del nostre model. El conjunt de tots els noms que ens permeten referenciar totes les realitzacions diferents d'aquest concepte, és a dir, el conjunt de tots els noms que ens permeten referenciar tots els clients l'expressem amb $NOMS_{Client}$.*

Noms del model i noms de la realitat. El conjunt de tots els noms de la realitat l'expressarem amb $NOMS_R$; el conjunt de tots els noms del model l'expressarem amb $NOMS_M$:

$$NOMS_R = \bigcup_{A \in REALITAT} NOMS_A \qquad NOMS_M = \bigcup_{A \in MODEL} NOMS_A$$

4.3 Propietats del llenguatge comú

4.3.1 Vides paral·leles

Un nom, dues lectures. Els noms usats en el llenguatge comú han tenir sentit tant en la realitat com en el model; un mateix nom ha de ser interpretable, *directament*,⁹ tant en termes del problema com de la solució. Per tant, els conjunts de noms de la realitat i del model han de coincidir:

$$NOMS_R = NOMS_M$$

Consistència denominadora. La doble interpretació d'un mateix nom, en el model i en la realitat, no és suficient per a la correcta comunicació. Per tal d'assegurar que la comunicació sigui efectiva cal que l'element del model expressat amb un nom sigui el model de l'element que el mateix nom expressa en la realitat: i anàlogament, que l'element de la realitat expressat amb un nom es modelitzi amb l'element que el mateix nom expressa en el model:¹⁰

$$nom_R(A) = nom_M(B) \Leftrightarrow model(A) = B$$

4.3.2 Ambigüitat denominadora

No ambigüitat. Cada nom només pot expressar un element en cadascun dels nivells de modelització (model o realitat), altrament la comunicació és ambigua. Suposem per exemple $nom_R(X) = nom_R(Y) = nom$, llavors quan usem *nom* per a la comunicació, a qui ens estem referint, a X o a Y?

⁹Aquí "directament" significa que la doble interpretació no requereix de cap mecanisme de modelització. A mesura que anem aprofundint en el tema de la comunicació caldrà matisar aquest "directament".

¹⁰Cal observar com aquesta afirmació només té sentit si hi ha isomorfia entre els dos mons. Però aquesta condició és una de les condicions necessàries per a la comunicació. \rightarrow

Ambigüitat denominadora en la realitat. Suposem $nom_R(X)=nom_R(Y)$, i suposem que existeix un element B del model tal que $nom_R(X)=nom_M(B)$. La *consistència denominadora* exigeix $model(X) = B$. Però per la igualtat dels noms en la realitat, també hem de tenir $model(Y) = B$. És a dir, la *consistència denominadora* juntament amb la presència d'*ambigüitat*, porta que un element del model té més d'un correlat en la realitat. I això impossibilita, en la comunicació, la distinció entre elements que en la realitat és pertinent.

Ambigüitat denominadora en el model. Suposem $nom_M(X)=nom_M(Y)$, i suposem que existeix un element A de la realitat tal que $nom_R(A)=nom_M(X)$. Per un argument similar a l'anterior, la *consistència denominadora* juntament amb la presència d'*ambigüitat*, porta que un element de la realitat té més d'un correlat en el model. I això impossibilita, en la comunicació, la distinció entre elements que en el model és pertinent.

Consistència i ambigüitat. Així, si no volem perdre les condicions necessàries per a la comunicació \rightarrow el que tenim és que la *consistència denominadora* i l'*ambigüitat denominadora* són incompatibles.

Conclusió 41, pàgina 263

(42)

Prohibició d'ambigüitat denominadora

La comunicació només és possible entre models i realitats en els que NO hi hagi *ambigüitat denominadora*

4.3.3 Anòmia

Elements sense nom. Si el mecanisme de comunicació és el nom de les coses, és evident que només es podran comunicar aquells elements que tinguin nom.

Definició. Anòmia

Direm que un model o una realitat és anòmica, quan conté elements sense nom

(43)

Prohibició d'anòmies

La comunicació només és possible entre models i realitats que NO són anòmics

4.3.4 Propietats del llenguatge comú de comunicació

Propietats del llenguatge comú de comunicació.

Les propietats que cal exigir al llenguatge de comunicació entre el model i la realitat són:

- **Sintaxi comuna.** És un conjunt de noms *NOMS*, comú tant en el model com en la realitat.
- **Semàntiques consistents.** La semàntica de cada nom és consistent en els dos mons:

$$nom_M(B) = nom_R(A) \Leftrightarrow B \in model(A)$$

- **Referent únic.** Tot nom expressa un sol element en cada un dels mons.
- **Completesa.** Tot element per comunicar té un nom, és a dir, en cap dels dos mons hi ha anòmies.

(44)

4.3.5 Condicions necessàries per a la comunicació

Estructura i llenguatge. Les condicions estructurals de la conclusió 41, pàgina 263, \rightarrow juntament amb les condicions sobre el llenguatge de comunicació, expressades en la conclusió 44, \rightarrow formen un conjunt de condicions necessàries i suficients per a la comunicació (perfecta).

Conclusió 41, pàgina 263

Conclusió 44, pàgina 267

Condicions suficients per a la comunicació. En l'apartat que segueix analitzem si l'ús dels principis de desenvolupament és suficient per assegurar les condicions necessàries per a la comunicació.

5 Comunicació i principis de modelització

5.1 Objectiu que perseguim

El paper dels principis en la comunicació. Els principis de modelització presentats faciliten el procés de modelització justament perquè faciliten la comunicació entre el model i la realitat. I això ho aconsegueixen gràcies a mantenir els dos mons amb estructura similar (*Espill*), a posar restriccions en els noms possibles en el model (*Franquícia* i *Franquícia obligada*), i a determinades exigències sobre els noms del model i de la realitat (*Referent únic*).

Suficiència dels principis de modelització. La qüestió fonamental, però, és si els principis de modelització són suficients per assegurar les propietats necessàries per a la comunicació perfecta.

? (37)

Els principis de modelització asseguruen les propietats necessàries per a la comunicació perfecta?

5.2 Models comunicables

Completesa i solidesa. Recordatori: La comunicació només té sentit davant de models *sòlids i complerts*.

Imperfeció en la comunicació. Recordatori: Aquesta comunicació és imperfecta per mor de la manca d'unicitat en la correspondència entre els elements del model i els de la realitat.

Perfeció en la comunicació. Recordatori: Per tal de poder tenir una comunicació perfecta cal exigir la *isomorfia* entre els dos mons a comunicar.

Coherència. Recordatori: si es compleixen totes les condicions anteriors diem que el model és *coherent* amb la realitat modelitzada.

El principi *Espill*. Recordatori: En un model construït segons el principi de l'*Espill*[→] hi ha una correspondència 1 – 1 entre els elements del model i els de la realitat; res però impedeix que hi hagi elements de la realitat que no tinguin cap correlat en el model.

1.1.1. Model i 'realitat', pàgina 8

(45)

Propietats assegurades per *Espill*

Per definició, el principi de l'*Espill*, si no es viola en cap moment, assegura les següents propietats:

1. **Solidesa.** Tot element del model té correlat en la realitat
2. **Isomorfia.** Tot element del model té un únic correlat en la realitat; els elements de la realitat tenen com a molt un correlat en el model

El que li manca a *Espill*. Així, el principi *Espill* proporciona quasi totes les condicions estructurals per assegurar la possibilitat de la comunicació entre el món i la realitat. Només manca assegurar la *completesa*.

Completesa. La *completesa*, però, és una propietat que cal exigir en tot procés de modelització: justament el model es construeix per “capturar” la realitat *pertinent*, i no pas necessàriament *tota* la realitat. No hi ha però cap principi concret que l'asseguri.

5.3 Llenguatge de noms i modelització

5.3.1 Noms d'un element

Multiplicitat de noms. Per tal de realitzar els raonaments és convenient considerar el cas en el que un mateix element tingui més d'un nom.

Definició. Sinonímia

La sinonímia significa que un element pot tenir més d'un nom; és a dir, el nom d'un element és de fet un conjunt de noms.

Canvis exigits per la multiplicitat dels noms. El fet de veure $nom(A)$ com un conjunt de noms, enlloc de veure'l com un sol nom, té algunes conseqüències sobre tot el dit fins ara sobre els noms, així com en la manera d'enunciar els principis de modelització. Enlloc d'entrar-hi ara, però, ho anirem fent paulatinament en els següents apartats.

5.3.2 Semàntiques consistents

Semàntiques consistents en termes planers. Sigui nom un nom usat en la realitat per fer referència a un element A , i en el model per fer referència a un element B . És a dir, $nom \in nom_R(A)$, i $nom \in nom_M(B)$.¹¹ El que diu la propietat de les *semàntiques consistents* és que aquesta coincidència de noms és possible perquè els elements als que fan referència estan lligats per la modelització, i a la inversa:

Semàntiques consistents:

$$nom_R(A) \cap nom_M(B) \neq \emptyset \Leftrightarrow B \in model(A)$$

Els principis de modelització asseguren la propietat de les semàntiques consistents sobre el conjunt de noms emprat?

(38) ?

Franquícia. La *Franquícia*, en presència de *sinonímia*, diu que si un dels noms del model de B coincideix amb un dels noms de la realitat de A llavors $model(A) = B$:

$$nom_M(B) \cap nom_R(A) \neq \emptyset \Rightarrow B \in model(A)$$

Franquícia obligada. El principi de la *Franquícia obligada* diu que en cas que en la realitat tinguem un nom que pot ser usat pel model, aquest l'ha d'usar. És a dir, durant la modelització el manlleu de noms, si és possible, es converteix en obligatori.

¹¹Cal observar com l'ús de conjunts de noms impedeix dir $nom_R(A) = nom_M(B) = nom$.

El significat de “poder usar”. Entenem que quan es parla de la possibilitat de manllevar un nom de la realitat és perquè es donen les condicions necessàries per a l’ús d’aquest nom en el model. Però aquestes condicions són les exigides pel principi de la *Franquícia*: cal que l’element B del model al qual se li aplica el nom manllevat, formi part del model de l’element A de la realitat del qual hem pres el nom.

Franquícia obligada i sinonímia. Així la *Franquícia obligada*, en presència de *sinonímia*, diu que en cas que un element formi part del model d’un altre (condició necessària pel manlleu del nom), cal que ambdós elements comparteixin algun nom:

$$B \in \text{model}(A) \Rightarrow \text{nom}_M(B) \cap \text{nom}_R(A) \neq \emptyset$$

(46)

Franquícia i consistència de semàntiques.

Un model construït segons els principis de la *Franquícia* i de la *Franquícia obligada* assegura la propietat de les *semàntiques consistentes* sobre el conjunt de noms.

5.3.3 Sintaxi comuna

Conjunt de noms. En l’anàlisi que segueix considerarem tres conjunts de noms:

- $NOMS_{M \cap R}$ o conjunt de noms compartit pel model i la realitat
- $NOMS_{M \setminus R}$ o conjunt de noms propis del model i que no apareixen la realitat
- $NOMS_{R \setminus M}$ o conjunt de noms propis de la realitat i que no apareixen en el model

Suficiència dels noms compartits. Tot seguit ens proposem demostrar que el conjunt $NOMS_{M \cap R}$ és suficient per a la comunicació. Això és, que tot element a comunicar, sigui del model o de la realitat, té un nom a $NOMS_{M \cap R}$.

Esbós de la demostració

1. Element no capturat. Suposem un element X en algun dels dos mons (model o realitat) que no és referenciable per cap dels noms compartits: $(\nexists \text{nom} \in NOMS_{M \cap R})(\text{nom} \in \text{nom}(X))$
2. Element no capturat del model.
 - (a) No solidesa. Suposem que X és un element del model; és a dir, $X \in \text{MODEL}$. Llavors afirmem que el model no és sòlid respecte la realitat modelitzada. Els següents passos ho demostren.

- (b) Supòsit de solidesa. Suposem que el model és sòlid, i que per tant $(\exists A \in \text{REALITAT})(X \in \text{model}(A))$
 - (c) Res d'anòmies. Assumim que no hi ha anòmies, i per tant $(\exists \text{nom})(\text{nom} \in \text{nom}_R(A))$
 - (d) Franquícia obligada. La *Franquícia obligada* exigeix el manlleu d'un nom: $(\exists \text{nom} \in \text{nom}_R(A))(\text{nom} \in \text{nom}_M(X))$. I el manlleu és possible ja que la no existència d'anòmies en la realitat ens diu que l'element modelitzat com a mínim té un nom.
 - (e) Contradicció. El resultat anterior significa que existeix un nom comú que ens serveix per referenciar X ; en d'altres paraules: $(\exists \text{nom} \in \text{NOMS}_{M \cap R})$. I això contradiu la hipòtesi inicial. Per tant concloem que, sota la hipòtesi de la *Franquícia obligada*, el model no és sòlid.
3. Element no capturat de la realitat.
- (a) No completa. Suposem que X és un element de la realitat; és a dir, $X \in \text{REALITAT}$. Llavors afirmem que el model considerat no és complet respecte aquesta realitat. Els següents passos ho demostren.
 - (b) Supòsit de completa. Suposem que el model és complet, i que per tant $(\exists B \in \text{MODEL})(B \in \text{model}(X))$
 - (c) Res d'anòmies. Assumim que no hi ha anòmies, i per tant $(\exists \text{nom})(\text{nom} \in \text{nom}_M(B))$
 - (d) Franquícia obligada. La *Franquícia obligada* exigeix el manlleu d'un nom: $(\exists \text{nom} \in \text{nom}_R(X))(\text{nom} \in \text{nom}_M(B))$. I el manlleu és possible ja que la no existència d'anòmies en el model ens diu que l'element B té com a mínim té un nom.
 - (e) Contradicció. El resultat anterior significa que existeix un nom comú que ens serveix per referenciar X : en d'altres paraules, $(\exists \text{nom} \in \text{NOMS}_{M \cap R})$. I això contradiu la hipòtesi inicial. Per tant concloem que, sota la hipòtesi de la *Franquícia obligada*, el model no és complet.

Suficiència dels noms comuns.

Donat un model *sòlid i complet*¹² sobre una determinada realitat, si aquest model segueix el principi de la *Franquícia obligada* llavors el conjunt de *noms comuns* permet referenciar tots i cadascun dels elements d'ambdós mons

(47)

¹²Aquí hem demostrat aquesta propietat pel cas en què ni el model ni la realitat són anòmiques. Però, tal i com veurem més endavant, \rightarrow l'ús de la *Franquícia obligada* impedeix l'existència d'anòmies en qualsevol dels dos mons. Per això no indiquem en el resultat que el model i la realitat ha de ser anòmiques.

Sintxi comuna. La conclusió anterior diu que si usem el conjunt de noms $NOMS_{M \cap R}$ tenim assegurada la propietat de *sintaxi comuna*, \rightarrow sempre i quan assegurem la completeness i solidesa del model, i construïm el model segons el principi de la *Franquícia obligada*.

Sintaxi comuna per a la comunicació. La completeness i solidesa són condicions estructurals que han de tenir els dos móns que volem comunicar; el llenguatge de noms és el llenguatge que emprarem per a la comunicació. És evident, però, que en usar el llenguatge de comunicació hem d'assumir que els mons són comunicables, altrament la comunicació no tindria sentit. Així, la conclusió anterior diu que si el model i la realitat tenen l'estructura necessària per a la comunicació, i usem la *Franquícia obligada*, llavors el llenguatge de noms comuns té la propietat de la *sintaxi comuna*.

(48)

Franquícia i sintaxi comuna.

En un model sòlid i complert¹³ construït segons la *Franquícia obligada* tenim assegurada la *Sintaxi comuna*

5.3.4 Referent únic

Recordatori: La propietat del Referent únic. La propietat del *Referent únic* \rightarrow diu que tot nom expressa un sol element en cadascun dels mons considerats. És a dir, que no hi ha *ambigüitat denominadora*: donat un nom sabem a quin element (únic) fa referència.

Recordatori: El principi del Referent únic. El principi del *Referent únic*, \rightarrow diu que en construir el model, els noms que donem als seus elements no poden generar ambigüitat referencial en cap dels dos mons considerats.

Propietat i principi. El *Referent únic* com a propietat diu com s'han de comportar els noms; el *Referent únic* com a principi diu com són els noms que podem usar en construir un model nou.

Referent únic en la realitat. La propietat del *Referent únic* aplicada a la realitat exigeix que tot nom de la realitat faci referència a un sol element d'aquesta. En cas de no ser així, abans de construir-ne un model o de pensar en un mecanisme de comunicació entre la realitat i el model, caldrà pensar en mecanismes de *desambiguació*.

El paper del glossari. En l'*especificació* el glossari és precisament el mecanisme de desambiguació previ a la modelització: el seu paper és aconseguir el *Referent únic* en la realitat que volem modelitzar.

¹³ Aquí hem demostrat aquesta propietat pel cas en què ni en model ni la realitat són anòmiques. Però, tal i com veurem mñes endavant \rightarrow l'ús de la *Franquícia obligada* impedeix l'existència d'anòmies en qualsevol dels dos mons. Per això no indiquem en el resultat que el model i la realitat ha de ser anòmiques.

Referent únic en el model. És important observar com ni l'ús de la *Franquícia obligada* ni el de la *Franquícia*, ni tant sols l'ús simultani dels dos principis, assegura la propietat del *referent únic* en el model. Ho mostrem en els següents exemples.

Exemple 67 (Violació del referent únic en el model) *Siguin dos elements X i Y del model. Suposem que $\text{nom}(X) = \{x, z\}$ i que $\text{nom}(Y) = \{y, z\}$.*

Siguin A i B dos elements de la realitat tals que $X \in \text{model}(A)$ i que $Y \in \text{model}(B)$. Suposem $\text{nom}(A) = \{x\}$ i $\text{nom}(B) = \{y\}$.

*En aquest cas es compleixen tant la *Franquícia* com la *Franquícia obligada*, i malgrat tot el nom z del model és ambigu.*

Exemple 68 (Violació del referent únic en la realitat) *Siguin dos elements A i B de la realitat. Suposem que $\text{nom}(A) = \{a, c\}$ i que $\text{nom}(B) = \{b, c\}$.*

Siguin X i Y dos elements del model tals que $X \in \text{model}(A)$ i que $Y \in \text{model}(B)$. Suposem $\text{nom}(X) = \{a\}$ i $\text{nom}(Y) = \{b\}$.

*En aquest cas es compleixen tant la *Franquícia* com la *Franquícia obligada*, i malgrat tot el nom c de la realitat és ambigu.*

Necessitat del principi del Referent únic. La presència del principi del *Referent únic* evita justament que es produeixi l'ambigüitat denominadora que mostren els exemples anteriors.¹⁴

5.3.5 Completesa del conjunt de noms

Recordatori: la *Franquícia obligada*. La *Franquícia obligada* diu que tot element del model que té correlat en la realitat ha de prendre el nom de la realitat que modelitza:

$$B \in \text{model}(A) \Rightarrow \text{nom}_M(B) \cap \text{nom}_R(A) \neq \emptyset$$

La *Franquícia obligada* prohibeix les anòmies en la realitat. Suposem que A sigui una anòmia en la realitat, i que A té correlat en el model. Llavors la *Franquícia obligada* no es podrà complir perquè és impossible que la intersecció amb un conjunt buit sigui un conjunt no buit. Per tant, si el model s'ha construït seguint el principi de la *Franquícia obligada* això significa que la realitat modelitzada no té anòmies.

La *Franquícia obligada* prohibeix les anòmies en el model. Suposem que B sigui una anomia en el model, i que B té un correlat en la realitat. Llavors la

¹⁴En el cas que no admetem la sinonímia els principis es poden relaxar. L'anàlisi corresponent, però, queda fora de l'abast dels nostres objectius.

Franquícia obligada no es podrà complir perquè és impossible que la intersecció amb un conjunt buit sigui un conjunt no buit. Per tant, si el model s'ha construït seguint el principi de la *Franquícia obligada* això significa que no té anòmies.

Completesa del conjunt de noms. La prohibició d'anòmies significa que cada element, del model o de la realitat, té com a mínim un nom. I aquesta és precisament la propietat de *completesa* exigida al llenguatge de comunicació. →

Conclusió 44, pàgina 267

5.4 Conclusions: principis de modelització i comunicació

5.4.1 Model comunicable

La responsabilitat del desenvolupador. La construcció d'un model que no viola *Espill* assegura la *solidesa del model* i la *isomorfia* entre els dos mons. És responsabilitat del desenvolupador assegurar-ne la *completesa*. El resultat és un model (*perfectament*) comunicable.

(49)

Espill i coherència

Un model *complet* que no viola *Espill* és un model *coherent* respecte la realitat modelitzada

5.4.2 Conjunt de noms vàlid per a la comunicació

Noms per a la comunicació. Sigui un model i una realitat, tal que tenen les condicions necessàries per a ser comunicables. El principi de la *Franquícia* limita els noms que poden tenir els elements del model (introdueix prohibicions); el principi de la *Franquícia obligada* exigeix que el model tingui uns noms determinats, tot i que deixa la porta oberta a la presència d'altres noms.¹⁵ Els noms útils de cara a la comunicació són els que són comuns en el món i en la realitat.

Principis necessaris per a la comunicació. Sigui un model i una realitat, tal que tenen les condicions necessàries per a ser comunicables. Si el model resultant s'ha construït sense violar la *Franquícia obligada* llavors el conjunt de *noms comuns* entre el model i la realitat, és una *sintaxi única* que a més és *completa*. Si a més el model s'ha construït sense violar la *Franquícia* llavors aconseguim la *consistència de les semàntiques*.

(50)

El llenguatge de comunicació¹⁶

El conjunt de *noms comuns* entre el model i la realitat d'un model *complet* que segueix els principis de l'*Espill*, la *Franquícia*, *Franquícia obli-*

¹⁵En concret, el principi de la *Franquícia obligada* demana una intersecció no buida entre els noms d'un element de la realitat i el noms de tot element que el modelitzi.

gada i el *Referent únic* és útil com a llenguatge de comunicació entre el model i la realitat.

6 La comunicació és possible

Conclusions planeres. Tot seguit plantejem les conclusions d'aquest capítol en termes planers. Entre parèntesis fem referència al concepte tècnic associat. Cal tenir present que hi ha condicions que cal exigir des d'un punt de vista tècnic, que no apareixen en l'exposició planera presentada.

Comunicació per noms. Per tal de comunicar el sistema amb el seu exterior, el problema amb la solució, el model amb la realitat, ens cal un llenguatge comú en ambdós mons. Aquest llenguatge és un conjunt de noms. (Sintaxi comuna).

Supressió dels elements sense nom. Això significa que els elements de la realitat o del model que no tinguin nom no podran ser comunicats. (Model i realitat no anòmics).

Sabem què vol dir cada nom. El significat dels noms en cada món ha de quedar ben clar. Per això en cal una definició explícita, clara i no ambigua. (Glossari).

Què comuniquem. De cara a la comunicació considerarem només aquells elements de la realitat amb nom que tenen correlat en el model, i aquells elements del model amb nom que tenen correlat en la realitat. Com a cas particular, només considerarem aquells *conceptes* amb nom que tenen correlat en el model, i aquells *components* amb nom que tenen correlat en la realitat.

Selecció dels noms. Els noms del model s'obtenen de la realitat modelitzada.¹⁷ Com a cas particular, els noms dels *components* s'obtenen dels noms dels *conceptes*. (*Franquícia, Franquícia obligada*).

¹⁶Si no admetem la sinonímia tot es pot simplificar una mica. Ho deixem com a exercici.

¹⁷De fet podem tenir noms en el model que no s'hagin obtingut de la realitat modelitzada. Però això no treu que per a la comunicació calen els noms manlevats de la realitat.

Principis i definicions del capítol

Definicions

Anòmia, 266

Completesa, 259

Isomorfia entre model i realitat, 262

Model coherent, 262

Nom, 264

Sinonímia, 269

Solidesa, 259

Capítol 15

Identificadors

1	Esbós del camí	282
2	Llenguatge per nivells	282
2.1	Mons a dos nivells	282
2.2	Múltiples models	284
2.3	Múltiples llenguatges	285
2.4	Llenguatge per nivells: resum	286
3	Propietat d'identificació	288
3.1	El paper de la propietat d'identificació	288
3.2	Identificació i propietat d'identificació	289
3.3	Identificació i noms	291
3.4	Extensió d'una propietat d'identificació	292
3.5	Propietat d'identificació especular	293
4	Component ben definit	296
4.1	Una definició un xic complexa	296
4.2	La unió fa la força	297
5	Identificació i llenguatge de comunicació	298
5.1	Associacions i visibilitats	299
5.2	Conceptes associatius	303
5.3	Especialitzacions	305
6	La comunicació és possible	313
	Definicions	315

Contingut detallat del capítol 15

1	Esbós del camí	282
	Encontres en la tercera fase	282
	Llenguatge per nivells	282
	Els identificadors	282
	Components ben definits	282
	Interrelacions entre components	282
2	Llenguatge per nivells	282
2.1	Mons a dos nivells	282
	Abstraccions i realitzacions	282
	Necessitat de l'abstracció i de la realització	282
	Comunicació doble	283
	L'abstracció com a element	283
	L'abstracció com a fragment	283
	Un món de dos nivells	283
2.2	Múltiples models	284
	Realitat amb dos nivells	284
	Model sobre les abstraccions	284
	Model sobre les realitzacions	284
	D'un a molts models	284
	Construcció del model corresponent a una abstracció	284
	Independència	285
	Model de totes les realitzacions	285
	Model abstracte i model concret: repetició i refinament	285
2.3	Múltiples llenguatges	285
	Necessitats de comunicació	285
	Llenguatge per nivells	285
	Llenguatge de noms i conjunt de noms	286
	Noms per a les abstraccions	286
2.4	Llenguatge per nivells: resum	286
2.4.1	Model sobre les abstraccions	286
	Mecanisme de construcció	286
	Model i realitat	286
	Llenguatge de les abstraccions	286
2.4.2	El llenguatge de comunicació	287
	Llenguatge sobre les abstraccions	287
	Llenguatge sobre les realitzacions	287
	Llenguatge de comunicació	287
3	Propietat d'identificació	288
3.1	El paper de la propietat d'identificació	288
	Llenguatges que cal construir	288
	Model i nom de les abstraccions	288
	Model dels components	288
	Problema en la construcció de $MODEL_B$	288

Una observació pertinent	288
Un camí per trobar la solució	288
Construcció de L_B	288
Individualització de realitzacions	288
La propietat d'identificació	289
Una mirada endavant	289
3.2 Identificació i propietat d'identificació	289
La solució dels identificadors	289
Individualització de la col·lectivitat	290
Identificador i propietat d'identificació	290
Ambigüïtat del terme <i>identificador</i>	290
Excursió. (Unicitat de l'identificador)	291
3.3 Identificació i noms	291
Universalitat de la identificació	291
Excursió. (Universalitat en la implementació: serialització)	292
3.4 Extensió d'una propietat d'identificació	292
Conjunt de noms	292
L'extensió d'una <i>propietat d'identificació</i>	292
Conjunt de propietats d'identificació	293
3.5 Propietat d'identificació especular	293
Propietat bàsica del model d'un conjunt d'entitats	294
La <i>identificació especular</i> en termes planers	294
4 Component ben definit	296
4.1 Una definició un xic complexa	296
El significat de <i>comunicable</i>	296
El significat de l' <i>estructura comunicable</i>	296
El significat de les <i>realitzacions identificables</i>	297
Excursió	297
4.2 La unió fa la força	297
El significat de la doble comunicabilitat	297
Components identificables	297
Llenguatge de comunicació	297
Excursió. (Identificació d'ús extern, intern, i doble)	298
5 Identificació i llenguatge de comunicació	298
Un model a dos nivells que queda coix	298
5.1 Associacions i visibilitats	299
5.1.1 Llenguatge dels enllaços	299
Context de l'anàlisi	299
Model d'una associació	299
Supòsit d'anàlisi	299
Excursió. (Múltiples models d'una associació)	299
Nom d'un enllaç	299
Nom d'un enllaç dirigit	299
Propietat d'identificació d'una associació	299
Propietat d'identificació d'una associació en termes del com- ponents	300

	Propietat d'identificació d'una visibilitat	300
	Comunicació d'enllaços	300
	Generalització	301
	Excursió. (Associacions amb pota a 1)	302
	Semàntica d'una associació	302
	Semàntica d'una visibilitat	302
	Llenguatges alternatius	302
5.1.2	Llenguatge de les associacions i visibilitats	303
	Comunicació d'enllaços	303
	La importància del context	303
	Coherència del MODEL	303
	Nom d'una associació	303
	Llenguatge de les abstraccions	303
5.2	Conceptes associatius	303
	Conceptes associatius i associacions	303
	Identificació d'un concepte associatiu ¹	303
	Conceptes associatius en el model	304
	Identificació del model d'un concepte associatiu	304
	Llenguatge de les realitzacions dels conceptes associatius	304
	Llenguatge dels conceptes associatius	305
5.3	Especialitzacions	305
5.3.1	Les realitzacions de les especialitzacions	305
	Model d'una especialització	305
	Nom de la realització d'una especialització	305
	Noms alternatius	306
5.3.2	Identificació per restricció de la generalització	307
	Restricció del domini	307
	Consistència de valors	307
	Restricció sobre un domini	307
	Excursió. (Propietat d'identificació com a relació)	307
	Identificació d'una especialització	308
5.3.3	Identificació independents	308
	Ús de propietats d'identificació independents	308
	Llenguatge per nivells	308
	El cas econòmic	308
	Pèrdua d'economia	309
	La independència no evita la dependència	309
	Factorització de la identificació	309
	Excursió. (Factorització de la identificació)	310
	Conclusions sobre la independència	310
5.3.4	Comunicació de les especialitzacions	310
	Context de l'anàlisi	310
	Objectiu que perseguim	310
	Model amb una especialització	310

¹Aquest resultat és l'obtingut amb les associacions, però adaptat als conceptes associatius.

Identificació de l'especialització introduïda a MODEL'	310
Construcció d'una especialització	311
Assegurar la comunicabilitat	311
Estructura de subconjunt	311
Identificació heretada	311
Excursió. (Especialització que és un concepte associatiu)	312
Excursió. (Especialització d'un un concepte associatiu)	312
6 La comunicació és possible	313
Conclusions planeres	313
Suposem que hem fet bé les coses	313
Mecanisme d'individualització	313
El nom de les realitzacions és útil	313
Què és un identificador	313
El paper dels identificadors	313
Interrelacions entre els components	313
Identificadors i models	313
Conclusió	313
Definicions	315

1 Esbós del camí

! (39)

Aquest capítol és fonamentalment tècnic. I per tant complex. En una primera lectura es pot passar per alt, a excepció del present apartat, on s'explica què es pretén i perquè, i del darrer, on s'exposen les conclusions en termes planers.

Encontres en la tercera fase. En el 14.[Encontres en la tercera fase](#), pàgina 253, hem analitzat les propietats estructurals que ha de tenir un model per tal que la comunicació sigui possible; les propietats que ha de tenir el llenguatge usat per a la comunicació; i com els principis de modelització assegurin les propietats necessàries i ens donen un llenguatge útil de cara a la comunicació.

Llenguatge per nivells. Sabem doncs que la comunicació és possible, i que el llenguatge de comunicació és el conjunt de noms comú en ambdós mons. Per tant cal assegurar que cada element de la realitat i cada element del model tinguin un nom. Aquesta és una tasca feixuga que a 15.2.[Llenguatge per nivells](#), pàgina 282, es simplifica: per comunicar una realització qualsevol ho fem per nivells, comuniquem per separat el nom de l'abstracció i el nom de la realització d'aquesta abstracció que estem considerant.

Els identificadors. Els *identificadors textuais*, que s'introdueixen a l'apartat 15.3.[Propietat d'identificació](#), pàgina 288, són el mecanisme que emprarem per donar nom a les realitzacions de les abstraccions. Els *identificadors textuais* asseguraran la coherència del model i constituïran la base del llenguatge de comunicació.

Components ben definits. En l'apartat 15.4.[Component ben definit](#), pàgina 296, es presenta la definició de *component ben definit*, que encapsula totes les propietats, estructurals i de nominació que han de complir els components del model per tal que la comunicació sigui possible.

Interrelacions entre components. Finalment, a l'apartat 15.5.[Identificació i llenguatge de comunicació](#), pàgina 298, analitzem com només amb la identificació sobre els components en tenim prou per comunicar les diferents interrelacions entre els components.

2 Llenguatge per nivells

2.1 Mons a dos nivells

Abstraccions i realitzacions. Tant les especificacions (món del problema) com els models (món de la solució) que manipulem consten d'*abstraccions* i *realitzacions*.

Necessitat de l'abstracció i de la realització. Considerarem que que tota realització només té sentit com a realització de l'abstracció que li és pròpia; i les abstraccions només tenen sentit si contenen alguna realització.

Comunicació doble. La consideració anterior fa que en cas que vulguem comunicar una realització també hem de poder comunicar l'abstracció; i en el cas que vulguem comunicar una abstracció també hem de poder comunicar les seves realitzacions.

Elements comunicables.

Els elements que volem comunicar són les *abstraccions* i les corresponents *realitzacions*.

(51)

L'abstracció com a element. Tota *abstracció* és un *element* del món considerat. Per exemple: els *components* són elements del MODEL; els *conceptes* són elements de la REALITAT.

L'abstracció com a fragment. Una *abstracció* la podem veure com un *fragment* del món considerat; en concret com el fragment format per totes les realitzacions d'aquesta abstracció. Així, les abstraccions són un *mecanisme de fragmentació* del món considerat.

Un món de dos nivells. La doble visió d'una *abstracció* com a *element* del món, i com a *fragment* del mateix món, permet veure aquest món com una estructura amb dos nivells d'abstracció: el nivell de les *abstraccions* i el nivell de les *realitzacions*.

Exemple 69 (Realitat de dos nivells) *El nostre model conceptual té dos conceptes: Client i Albarà. (Pel que fa a l'exemple ens podem oblidar de les associacions presents en el model).*

En semprePaga és un bon client, mentre que en pereBerenguera és un client força recent. Els albarans els anem numerant consecutivament, i de moment en tenim cinc.

La nostra realitat consta per tant de cinc entitats: REALITAT = {semprePaga, pereBerenguera, 1, 2, 3, 4, 5}²

Aquesta realitat però la podem expressar com una realitat formada per dues abstraccions: REALITAT = {Client, Albarà}.

El Client és un fragment de la realitat que conté dues entitats: Client={semprePaga, pereBerenguera}.

L'{Albarà} és un altre fragment de la realitat que conté cinc entitats: Albarà={1,2,3,4,5}.

²Expressem cada entitat amb un valor que la identifica.

(52)

Fragments de comunicació.

Una abstracció es pot veure com un fragment del món al que pertany. Per tant, cal assegurar que la comunicació sigui *possible* dins dels fragments expressats per les diferents abstraccions.

2.2 Múltiples models

Realitat amb dos nivells. Partim d'una realitat fragmentada en abstraccions, és a dir, estructurada en dos nivells d'abstracció.

Model sobre les abstraccions. En el capítol 14, *Encontres en la tercera fase*, pàgina 253, [→] hem vist que si construïm un model sobre les abstraccions de la realitat pertinent seguint els principis de modelització, haurem obtingut un model *coherent* amb la realitat modelitzada i un llenguatge vàlid per a la comunicació (de les abstraccions). [→]

14.Encontres en la tercera fase, pàgina 253

Conclusió 50, pàgina 274



Notació. Al model sobre les abstraccions de la realitat l'expressem amb $MODEL_{Abs}$.

Model sobre les realitzacions. Cada abstracció del model és també un model d'un fragment de la realitat. És a dir, *el model sobre les realitzacions que volem construir és la unió de totes les abstraccions del model vistes com a models*.

D'un a molts models. Així, tenim un model sobre les abstraccions de la realitat (el model abstracte $MODEL_{Abs}$), i cada element B d'aquest model és un model sobre les realitzacions $a:A$ de la realitat que són realitzacions de l'abstracció A modelitzada per B . Tenim doncs tants models sobre realitzacions com abstraccions tingui el model.

Construcció del model corresponent a una abstracció. Per cada abstracció B del model sobre les abstraccions, $MODEL_{Abs}$, construïm, seguint els principis de modelització, el model sobre les realitzacions de A (on A és la realitat modelitzada per B , $model(A) = B$). La conclusió 50, pàgina 274, [→] ens assegura que haurem obtingut un conjunt de models *coherents* amb la realitat modelitzada i un conjunt de llenguatges vàlids per a la comunicació de realitzacions (de B o de A). Cal observar que cada llenguatge només és vàlid pel fragment pertinent.

Conclusió 50, pàgina 274

Notació. El model sobre les realitzacions d'una abstracció A l'expressem amb MODEL_A . Si $\text{model}(A) = B$ tant podem usar MODEL_A com MODEL_B , i generalment usarem MODEL_B .³



Independència. La construcció de cada model MODEL_B és independent tant de la construcció dels altres models MODEL_B , com de la construcció del model MODEL_{Abs} sobre les abstraccions.

Model de totes les realitzacions. La unió dels models MODEL_B , on B és una abstracció present en el model MODEL_{Abs} , l'anomenem *model de totes les realitzacions* relatives a MODEL_{Abs} , i l'expressem amb $\text{MODEL}_{Conc}(\text{MODEL}_{Abs})$.⁴

$$\text{MODEL}_{Conc}(\text{MODEL}_{Abs}) = \bigcup_{B \in \text{MODEL}_{Abs}} \text{MODEL}_B$$

Model abstracte i model concret: repetició i refinament. El resultat de tot plegat és un model que de fet són dos models: un *model abstracte* i un *model concret*. El *model abstracte* és el model de les abstraccions de la realitat MODEL_{Abs} ; el *model concret* és el model de les realitzacions de la realitat relatives al *model abstracte*: $\text{MODEL}_{Conc}(\text{MODEL}_{Abs})$.

2.3 Múltiples llenguatges

Necessitats de comunicació. Acabem de veure com la distinció entre *abstraccions* i *realitzacions* fa que els mons que volem comunicar són mons estructurats en dos nivells. Per tant podem també introduir la comunicació a dos nivells: en qualsevol moment, per comunicar un element, caldrà comunicar primer de quin fragment del món estem parlant (comuniqueu l'abstracció) i tot seguit de quin element dins d'aquest fragment.

Llenguatge per nivells. La proposta d'usar dos noms per a les realitzacions, un per l'abstracció pertinent i un altre per la realització en qüestió dins l'abstracció, és una proposta de *llenguatge per nivells*. En concret:

- *Nivell de les abstraccions.* El llenguatge ha permetre comunicar el nivell d'abstracció. Ho fem amb els noms de les abstraccions.

³Aquesta notació permet dues lectures. MODEL_A es pot llegir com el model que captura totes les realitzacions de l'abstracció A de la realitat. En canvi, MODEL_B cal llegir-ho com el fragment del model que conté totes les realitzacions de l'abstracció B del model; el que passa és que aquestes realitzacions són totes el model d'una realització de l'abstracció A de la realitat.

Generalment usarem MODEL_B per emfatitzar que ens movem en l'àmbit del model, i que el model a què ens referim és el fragment que "encapsulem" en l'abstracció B del model.

⁴El subíndex Conc significa *concrecions*, en contraposició a *abstraccions*.

- *Nivell de les realitzacions.* La comunicació d'una realització es fa per nivells: primer es comunica l'abstracció, i un cop coneguda aquesta s'indica la realització dins d'aquesta abstracció.

Llenguatge de noms i conjunt de noms. El *llenguatge de noms* és aquell que és útil per a la comunicació, i que per tant és comú en el model i en la realitat. El *conjunt de noms del model* és el conjunt format per tots els noms usats en el model, i en general és diferent del *conjunt de noms de la realitat*.

Noms per a les abstraccions. L'ús d'un *llenguatge a dos nivells* significa que els noms de les realitzacions són de fet un parell $\langle \text{nomAbstracció}, \text{nomRealització} \rangle$, on cada *nomRealització* només té sentit dins de l'abstracció expressada per *nomAbstracció*.

(53)

Llenguatge a dos nivells⁵

- **Abstraccions.** Cada abstracció té el seu nom propi *nomAbstracció*
- **Realitzacions.** El nom d'una realització és el parell $\langle \text{nomAbstracció}, \text{nomRealització} \rangle$ on cada *nomRealització* només té sentit dins de l'abstracció expressada per *nomAbstracció*

2.4 Llenguatge per nivells: resum

2.4.1 Model sobre les abstraccions

Mecanisme de construcció. Usem els principis de modelització (*Espill*, *Franquícia*, *Franquícia obligada* i *Referent únic*) per a construir el model MODEL_{Abs} sobre les abstraccions.

Model i realitat. El model MODEL_{Abs} resultant és un model *coherent* sobre la *realitat*.⁶

Llenguatge de les abstraccions. La coherència del model MODEL_{Abs} , juntament amb el fet que s'ha construït seguint els principis de modelització, significa que el conjunt de noms definit per MODEL_{Abs} , que anomenem llenguatge L_{Abs} sobre les abstraccions, és un llenguatge vàlid per a la comunicació de les abstraccions.

⁵Per simplicitat expremem la conclusió en termes d'absència de sinonímia. La generalització pel cas de permetre la sinonímia és ben simple: n'hi ha prou en demanar que *nomAbstracció* i *nomRealització* siguin conjunts.

⁶Pròpiament cal assegurar la *completesa* del model. Però si ens limitem a aquells elements de la realitat que sí que tenen correlat en el model, llavors el model és *complet* per a aquest fragment.

Notació. El llenguatge de comunicació per a les abstraccions d'un model coherent l'expressem amb L_{Abs} .⁷



2.4.2 El llenguatge de comunicació

Llenguatge sobre les abstraccions. Donat el model $MODEL_{Abs}$ sobre les abstraccions, els noms d'aquest model que s'han manllevat de la realitat constitueixen el llenguatge L_{Abs} sobre les abstraccions.

Llenguatge sobre les realitzacions. Donat el model $MODEL_B$, on B és un component, els noms d'aquest model que s'han manllevat de la realitat constitueixen el llenguatge L_B sobre les realitzacions de B .

Notació. Donat un component B el llenguatge de les seves realitzacions l'anomenem L_B .



Llenguatge de comunicació. El llenguatge de comunicació és el llenguatge a dos nivells construït amb els llenguatges L_{Abs} i els múltiples llenguatges L_B , on B és una abstracció del model.

Construcció d'un model a dos nivells i d'un llenguatge par a la comunicació

1. **Descripció de la realitat modelitzada.** El *glossari* delimita la realitat modelitzada, i assegura la propietat del *Referent únic* en la realitat.
2. **Model sobre les abstraccions.** El construïm usant els principis de modelització (*Espill*, *Franquícia*, *Franquícia obligada* i *Referent únic*).
3. **Model sobre les realitzacions.** Per cada una de les abstraccions A de la realitat construïm un model $MODEL_B$ coherent.
4. **Llenguatge sobre les abstraccions.** El llenguatge L_{Abs} sobre les abstraccions és el conjunt de noms comuns entre $MODEL_{Abs}$ i $REAL_{Abs}$.
5. **Llenguatge sobre les realitzacions.** El llenguatge L_B de les realitzacions de cada component B és el conjunt de noms comuns entre $MODEL_B$ i $REAL_B$.
6. **Comunicació.** Per comunicar una abstracció usem el nom de $MODEL_{Abs}$; per comunicar una realització

(54)

⁷Si el model és coherent és comunicable, i per tant el llenguatge de comunicació és comú en el model i en la realitat. Per tant L_{Abs} també és el llenguatge de les abstraccions de la realitat modelitzada.

usem el nom pertient de $MODEL_B$ juntament amb el nom donat a B dins de $MODEL_{Abs}$.

3 Propietat d'identificació

3.1 El paper de la propietat d'identificació

Llenguatges que cal construir. El mecanisme de comunicació a dos nivells \rightarrow demana construir el llenguatge L_{Abs} del model considerat, i els llenguatges L_B per cada *component* B d'aquest model.

Model i nom de les abstraccions. Com hem vist, \rightarrow la construcció del model sobre les abstraccions seguint els principis de modelització ens proporciona el llenguatge L_{Abs} de comunicació.

Model dels components. El següent pas en la construcció del model a dos nivells i del corresponent llenguatge de comunicació \rightarrow és la construcció d'un model coherent $MODEL_B$ per cadascuna de les abstraccions de $MODEL_{Abs}$.

Problema en la construcció de $MODEL_B$. A diferència de $MODEL_{Abs}$, el model $MODEL_B$ generalment té un nombre infinit d'elements. I això dificulta tant assegurar-ne la coherència, com construir-lo segons els principis de modelització.⁸

Una observació pertinent. Els principis de modelització asseguren tant l'estructura del model (la coherència) com que els noms comuns formen un llenguatge de comunicació. I de fet, qui porta la batuta és el principi de la *Franquícia obligada*. És a dir, les restriccions donades en els noms del model ajuden tant a restringir l'estructura del model com a construir el llenguatge de comunicació.

Un camí per trobar la solució. L'observació anterior ens porta a pensar que l'anàlisi de les propietats de L_B ens pot ajudar a definir com es construeix $MODEL_B$.

Construcció de L_B . Donat un component B , la construcció del llenguatge L_B de les seves realitzacions significa donar un nom diferent a cadascuna de les realitzacions que constitueixen l'abstracció B . Cal però assegurar que no hi ha dues realitzacions amb el mateix nom (absència d'*ambigüitat referencial*), i que cada realització té com a mínim un nom (absència d'*anòmies*).

⁸Sabem \rightarrow que si un model es construeix segons els principis de modelització, i en podem assegurar la completesa, és un model coherent. I a més el llenguatge de noms comú en el model i en la realitat és un llenguatge vàlid per a la comunicació.

Individualització de realitzacions. En conclusió, *el nom d'una realització és un mecanisme individualitzador de la realització concreta dins del conjunt de totes les realitzacions de la mateixa abstracció.*

Els noms del model⁹ són un *mecanisme individualitzador* de les realitzacions (dins l'abstracció pertinent).

(55)

La propietat d'identificació. Enlloc de donar nom explícitament a cadascuna de les realitzacions, el que farem serà introduir una propietat, la *Propietat d'identificació*, capaç de donar a cada nova potencial realització un nom que mantingui les propietats de la comunicació perfecta. La presentació d'aquesta propietat és l'objectiu dels següent apartat.

Una mirada endavant. La propietat d'identificació ens permetrà donar nom a cadascuna de les realitzacions d'una abstracció. Amb això ja podrem definir quines propietats ha de tenir el model d'una abstracció de la realitat, i en concret quines propietats ha de tenir un component que no violi *Espill*. Finalment analitzarem fins a quin punt els principis de modelització asseguruen el que necessitem.

3.2 Identificació i propietat d'identificació

La solució dels identificadors. Els *identificadors* són el mecanisme més simple i habitual per resoldre la col·lisió comunicativa entre el món exterior i el món interior al sistema.

Definició. Propietat d'identificació

Propietat¹⁰ d'un conjunt de realitzacions, anomenat espai d'identificació, tal que:

1. **Obligatorietat.** *Tota element de l'espai d'identificació té com a mínim un valor per a aquesta propietat¹¹*
2. **Unicitat.** *No hi ha dues realitzacions diferents dins l'espai d'identificació que tinguin un mateix valor per a aquesta propietat¹²*

⁹Pròpiament, els noms que en el model donem a les realitzacions.

¹⁰Pròpiament l'identificador pot ser un conjunt de propietats. La definició en termes de propietat i no en termes de conjunt de propietats és força més simple i no implica pèrdua de generalitat.

¹¹Fixem-nos que estem admetent que per a aquesta propietat l'element tingui més d'un valor.

¹²Formalment, sigui ID_a el conjunt de valors que una realització $a:A$ té per a la propietat d'identificació. Llavors, si prenem un altra realització $b:A$ qualsevol hem de tenir $ID_a \cap ID_b = \emptyset$.

Definició. Espai d'identificació

Anomenem espai d'identificació al conjunt de realitzacions sobre el que es defineix una propietat d'identificació

Individualització de la col·lectivitat. La idea de la *propietat d'identificació* és que n'hi ha prou en conèixer el seu valor per saber de quina realització estem parlant dins del context de l'*espai d'identificació*. El valor de la *propietat d'identificació* és, per tant, un mecanisme individualitzador de la col·lectivitat representada per l'element.

Definició. Identificador d'una realització

Valor concret que la realització té de la propietat d'identificació.

Identificador i propietat d'identificació. L'*identificador d'una realització* és la realització de la *propietat d'identificació* corresponent.¹³ Una *propietat d'identificació* expressa quina propietat cal conèixer per individualitzar una realització en el conjunt de les altres realitzacions de l'*espai d'identificació*; l'*identificador d'una realització* determinada és el valor d'aquesta propietat que fa única la realització de l'element dins de l'*espai d'identificació*.

(56)

- La *propietat d'identificació* és una *abstracció*
- L'*identificador d'una realització* és la *concrecció* de l'abstracció representada per la *propietat d'identificació* corresponent

Exemple 70 (Identificador) El component *Caminada* té una propietat d'identificació anomenada *nom-Caminada*. És a dir, l'*espai d'identificació* considerat és el definit pel component *Caminada*, i per tant és el conjunt de totes les caminades del sistema.

La *c : Caminada* que en Pere i l'Anna van fer la setmana passada és la que té el valor *Cingles-de-Bertí* en aquesta propietat. És a dir, l'*identificador* de la realització en qüestió és *Cingles-de-Bertí*.

¹³Aquest redactat és altament simplificador. Res impedeix que donat un mateix espai d'identificació hi hagi més d'una propietat d'identificació; i res impedeix tampoc que una mateixa realització pertanyi a dos espais d'identificació diferents. En qualsevol d'aquests casos donada una realització caldrà explicitar de quina de les múltiples propietats d'identificació en les que participa estem parlant. A més, segons la definició, donada una propietat d'identificació una realització pot tenir més d'un valor, i per tant més d'un identificador. En conseqüència parlar de l'*identificador* d'una realització és d'una simplificació enorme.

Ambigüitat del terme *identificador*. Sovint usem *identificador* tant per referir-nos a l'*identificador d'una realització* com per referir-nos a la *propietat d'identificació*. Per desfer l'ambigüitat sovint usarem *valor de l'identificador* per referir-nos a l'*identificador d'una realització*.

Identificador és un mot ambigu (desambiguat sempre pel context) que tant pot referir-se a la *propietat d'identificació* com a l'*identificador d'una realització*.

(40) !

Excursió. (Unicitat de l'identificador) Generalment, i en contra de com ho hem presentat aquí, la *unicitat* de l'identificador es defineix en ambdós sentits: cada valor de la propietat d'identificació expressa una única realització; i a cada realització li correspon un únic valor per a aquesta propietat. En aquesta visió tradicional hi ha tantes realitzacions com valors de la propietat d'identificació.

La definició presentada aquí considera la *unicitat* en un sol sentit: cada valor de la propietat d'identificació expressa una única realització. D'aquesta manera deixem la porta oberta a la *sinonímia*: tot i que una mateixa realització pot ser expressada amb diferents valors de la propietat d'identificació, el que és fonamental és que cada valor expressi de manera unívoca una sola realització.

Notació. Per expressar que l'*espai d'identificació* d'una *propietat d'identificació* P és el conjunt x , direm que P és una *propietat d'identificació sobre* X .



3.3 Identificació i noms

Definició. Propietat d'identificació textual

Direm que una propietat d'identificació és textual si els valors d'aquesta propietat són sempre noms.

Exemple 71 (Identificador textual) Tots els identificadors usats en l'exemple 70 \rightarrow són noms. Això és així perquè la *propietat d'identificació nom-caminada* és una *propietat d'identificació textual*.

Exemple 70. Identificador, pàgina 290

Universalitat de la identificació. L'ús de *noms* com a *valors* de la *propietat d'identificació* permeten que aquests valors (o *identificadors*) puguin viure fora del món on "viuen" les realitzacions identificades per ells. És a dir, el *nom* com a *valor d'identificació* pot traspasar les fronteres del món on té la seva raó de ser.

Excursió. (Universalitat en la implementació: serialització) L'ús dels *noms* permet usar un mateix valor en qualsevol món possible; en concret, dins en el sistema, que és un món possible, però també a l'exterior del sistema, que és un altre món possible.

Un mecanisme per donar *nom* a qualsevol objecte és la *serialització*: es tracta d'expressar l'*estat intern* de la realització considerada en termes de seqüència de caràcters. Per *estat intern* entenem el conjunt ordenat dels valors dels atributs de l'objecte en qüestió.

Si bé tot objecte és serialitzable, no necessàriament la cadena resultant de la serialització és només vàlida per a un sol objecte. Per tant, la serialització no és suficient per obtenir una propietat d'identificació.

Exemple 72 (Serialització i identificació) Suposem un component *Vehicle* que té visibilitat monoavaluada d'atribut sobre un component *Coordenada*. La semàntica d'aquesta visibilitat és emmagatzemar la posició dels vehicles.

Donats dos *Vehicles*, *a* i *b*, cadascun d'ells té un enllaç dirigit sobre la seva pròpia *Coordenada*. Així, *a.coord* i *b.coord* són dos objectes *Coordenada* diferents. Res però impedeix que en un moment determinat els dos vehicles estiguin en la mateixa posició.¹⁴

Suposem doncs que tot i que *a.coord* i *b.coord* són diferents, tenen el mateix valor. Llavors la serialització de *a.coord* i la serialització de *b.coord* donaran la mateixa seqüència.¹⁵

! (41)

Mentre no es digui el contrari, en tot el que segueix, quan parlem de *propietat d'identificació* ens referim a una *propietat d'identificació textual*.

3.4 Extensió d'una propietat d'identificació¹⁶

Conjunt de noms. El conjunt de tots els *valors* possibles de la *propietat d'identificació*¹⁷ sobre un determinat espai d'identificació és el *conjunt de tots els noms* possibles que poden tenir les realitzacions d'aquest espai: cada realització té com a mínim un nom (*obligatorietat*); cada nom és per a una sola realització (*unicitat*).

¹⁴D'acord: o bé han xocat, ho bé hem d'admetre que el sistema de coordenades té un gra massa gruixut, de tal manera que enlloc d'expressar punts expressa superfícies més o menys amples!

¹⁵Això és així a no ser que un dels atributs de *Coord* sigui un identificador. Però llavors el mecanisme individualitzador de la coordenada és aquest identificador, i no pas la serialització de l'objecte.

¹⁶Restringim l'exposició als identificadors textuais per comoditat expositiva. Tanmateix tant les definicions com els resultats es poden expressar sense problemes en termes d'identificadors generals, no necessàriament textuais.

¹⁷Recordem que només considerem propietats d'identificació textuais, i que per tant els valors d'identificació sempre són noms.

L'extensió d'una propietat d'identificació. Tota propietat d'identificació defineix un *conjunt de noms*: l'extensió d'aquesta propietat. Tot *conjunt de noms* suficientment gran pot usar-se per definir una *propietat d'identificació*: aquella que la seva extensió és el conjunt donat.

Notació. Usarem $X \in \text{PROP}_{Id}(A)$ per expressar que X és una propietat d'identificació sobre A . Admetrem que X sigui una propietat (intensió) o un conjunt de noms (extensió).



Conjunt de propietats d'identificació. Pròpiament $\text{PROP}_{Id}(A)$ és el conjunt de totes les propietats d'identificació de A . Així, si X és un conjunt de noms, llavors $X \in \text{PROP}_{Id}(A)$ expressa que aquest conjunt de noms és una propietat d'identificació de A .

Notació. Sovint no ens interessa donar cap nom a les propietats d'identificació. En aquests casos usarem $\text{Prop}_{Id}(A)$ per expressar que tenim una propietat d'identificació sobre A , que anomenem $\text{Prop}_{Id}(A)$. És a dir, $\text{Prop}_{Id}(A) \in \text{PROP}_{Id}(A)$.



Notació. Per expressar que un dels valors qualsevol d'identificació d'una realització a usarem $id(a)$.¹⁸ En alguns casos caldrà indicar explícitament quina és la propietat d'identificació emprada, i ho farem en forma de subíndex. Per exemple, $id_{\text{Prop}}(a)$ o $id_{\text{Prop}_{Id}(A)}(a)$.



Notació. La definició de la propietat d'identificació donada, que permet valors diferents sobre una mateixa realització (és a dir, admet la sinonímia), fa que $id(a)$ pugui ser un conjunt. El context haurà de desambiguar quan amb $id(a)$ expressem un valor o un conjunt de valors.



3.5 Propietat d'identificació especular

Restringim l'exposició als identificadors textuais per comoditat expositiva. Tanmateix tant les definicions com els resultats es poden expressar sense problemes en termes d'identificadors generals, no necessàriament textuais.

(42) !

¹⁸Recordem que la definició de propietat d'identificació admet que una realització tingui més d'un valor per a aquesta propietat.

Definició. Model d'un conjunt d'entitats

Sigui $model(A) = B$, i sigui X un conjunt d'entitats de A : $X \subseteq A$. Definim el model d'aquest subconjunt d'entitats com el subconjunt d'objectes realització de B que són el model de les entitats de X :

$$model(X \subseteq A) = \bigcup_{a \in X} model(a)$$

Propietat bàsica del model d'un conjunt d'entitats. La definició de *model* d'un conjunt d'entitats parteix d'un subconjunt de les entitats que realitzen un determinat concepte. En aplicar la definició de model sobre aquest conjunt el que obtenim és un subconjunt del component que modelitza el concepte inicial:

$$model(X \subseteq A) \subseteq model(A)$$

Definició. Propietat d'identificació especular

*Direm que una propietat d'identificació X definida sobre un espai d'identificació B de realitzacions del model és una propietat d'identificació especular si i només si:*¹⁹

- **Identificació bivalent.** *També és una propietat d'identificació del conjunt A de realitzacions de la realitat que B modelitza, $B = model(A)$: $X \in PROP_{Id}(B) \Rightarrow X \in PROP_{Id}(A)$.*
- **Semàntiques consistents.** *Les realitzacions del model i la realitat que comparteixen valor d'identificació és perquè una és model de l'altra:*

$$(\forall b : B)(\forall a : A) id(b : B) \cap id(a : A) \neq \emptyset \Leftrightarrow b : B \in model(a : A)$$

La identificació especular en termes planers. Una propietat d'identificació especular és aquella en la que el mateix conjunt de noms ens és útil tant per identificar una realització del model com per a identificar la corresponent realització de la realitat: un mateix conjunt de noms ens serveix a les dues bandes del mirall.

¹⁹En aquesta definició partim d'una identificació en el model. I demanem que també ho sigui de la realitat. És evident que també podríem partir d'una identificació en la realitat, i demanar que també ho sigui del model. Però no ho fem així per dues raons:

- És en construir el model que posem noms. De fet abans de construir el model és possible que ni tant sols ens haguéssim plantejat que les abstraccions i les realitzacions de la realitat tenen nom. És més, el bateig per sí mateix ja és una tasca de modelització. Per exemple, si diem blanc, i no pas blanc₁ i blanc₂ és perquè en català en el moment de posar noms als colors no hem considerat que el nostre model lingüístic hagi de diferenciar entre dues tonalitats diferents del que considerem un mateix color; en canvi, d'altres llengües ho veuen diferent.
- Els models que considerarem són models comunicables, i per tant coherents. Els dos mons són isomorfs i per tant si podem identificar amb noms en un món, els mateixos noms ens serveixen per identificar en l'altre món.

Exemple 73 (Identificació especular) Suposem $\text{model}(A) = B$. Sigui $A = \{a1, a2\}$ amb $\text{nom}(a1) = \{n1\}$ i $\text{nom}(a2) = \{n2, n3\}$. Sigui $B = \{b1, b2\}$ amb $\text{nom}(b1) = \{n1, n2\}$ i $\text{nom}(b2) = \{n3\}$. Sigui $\text{model}(a1) = \{b1\}$ i $\text{model}(a2) = \{b1, b2\}$.²⁰ Llavors el conjunt de noms de B és un identificador especular; i l'espai d'identificació és B (i per tant també el concepte A).

L'identificació és bivalent ja que el mateix conjunt de noms permet tant identificar les realitzacions de B com les de A . És a dir, el mateix conjunt de noms tant és una propietat d'identificació definida sobre el conjunt d'objecte de B , com una propietat d'identificació definida sobre el conjunt d'entitats que el conjunt B modelitza.

Per a la consistència en les semàntiques caldrà analitzar totes les realitzacions. Ho fem per un parell i deixem les altres com a exercici.

La realització $b1 : B$ té dos valors d'identificació. El valor $n1$ identifica la realització $a1 : A$ de la realitat, mentre que el valor $n2$ identifica la realització $a2 : A$ de la realitat; en ambdós casos les realitzacions identificades tenen una relació de modelització amb $b1 : B$. Per tant es compleix la condició de les semàntiques consistents pel que fa $b1 : B$.

La realització $a1 : A$ té un sol valor d'identificació que també identifica la realització $b1 : B$ del model, que manté una relació de modelització amb $a1 : A$. Per tant es compleix la condició de les semàntiques consistents pel que fa $a1 : A$.

Manca comprovar la condició de les semàntiques consistents per a $a2 : A$ i per a $b2 : B$.

Si $\text{nom}2 \in \text{nom}(a1 : A)$ llavors la identificació deixa de ser especular perquè perd la condició de la bivalència: hi ha un valor d'identificació a B que a A no identifica per ser ambigu referencialment.

És important observar com tot i que el conjunt de noms de B és una propietat especular, no ens és útil de cara a la comunicació: els noms que en un món són sinònims (com per exemple $n1$ i $n2$ en el model), en l'altre món poden no ser-ho.

Exemple 74 (Identificació no especular) Suposem les mateixes condicions de l'exemple 73. \rightarrow

Si $\text{nom}3 \in \text{nom}(a1 : A)$ llavors la identificació deixa de ser especular perquè perd la condició de les semàntiques consistents: hi ha un valor d'identificació que a A expressa un element que la relació de modelització no lliga amb l'element de B que usa aquest mateix valor d'identificació. (En aquest cas, com que $\text{nom}3 \in \text{nom}(a2 : A)$ també perdem la bivalència: hi ha un valor d'identificació a B que a A no identifica per ser ambigu referencialment).

Si $\text{nom}4 \in \text{nom}(b1 : B)$ llavors la identificació deixa de ser especular

²⁰Compte, el model no és coherent, i per tant no és comunicable.

perquè perd la condició de la bivalència: hi ha un valor d'identificació a B que a A no identifica

Si $\text{nom4} \in \text{nom}(a1:A)$ llavors la identificació es manté especular. La bivalència es manté ja que tota propietat d'identificació de B també ho és de A . La consistència en les semàntiques es manté perquè el nom afegit no coincideix amb cap nom de B .



Notació. Direm que un *identificador* és *especular* si és el valor d'una propietat d'identificació especular.

4 Component ben definit

4.1 Una definició un xic complexa

Definició. Component ben definit

Donat un component B dins d'un model MODEL direm que està ben definit si i només si es compleix:

- **Model comunicable.** $\exists A,^{21}$ únic, tal que $\text{model}(A) = B,^{22}$
- **Estructura comunicable.** El conjunt de les realitzacions de B és un model sòlid i complert sobre el conjunt de les realitzacions de A .
- **Realitzacions identificables.** Existeix una propietat d'identificació PROP coneguda sobre B que n'és una propietat d'identificació especular.

El significat de comunicable. L'extensió de la propietat *comunicable* a tots els components del model MODEL assegura que el model (si els únics elements que cal considerar són els conceptes i els components) té les propietats estructurals necessàries per a la comunicació perfecta. \rightarrow És a dir, que el model MODEL sobre les abstraccions és un model *coherent* sobre les abstraccions de la realitat modelitzada.

El significat de l'estructura comunicable. Per tal que el component com a model compleixi les propietats estructurals necessàries per a la comunicació perfecta, \rightarrow cal exigir que sigui un model *coherent*. Tanmateix en la definició

²¹L'àmbit dels quantificadors, implícits i explícits, és el model MODEL , i la realitat REALITAT modelitzada.

²²La igualtat diu que el model no és un conjunt de components, i que per tant si anem de la realitat al model des de A el camí és únic. L'exigència d'unicitat de A ens diu que si fem el casmí invers, del model a la realitat des de B , el camí també és únic. Fixem-nos que si aquesta propietat l'extenem a tots els components aconseguim la isomorfia, almenys si només considerem els elements de la realitat que tenen correlat en el model.

15.4 Component ben definit

J.M. Merenciano

de *component ben definit* només s'exigeix la *solidesa* i la *completesa* del model. El motiu és que el que manca per a la *coherència* (la *isomorfia*: que cada element de la realitat tingui un correlat en el model, i a la inversa) és conseqüència de les condicions exigides a la propietat d'identificació Prop .

El significat de les realitzacions identificables. La propietat de les *realitzacions identificables* diu que l'espai d'identificació és el component \mathbf{B} , i que els noms del model permeten identificar tant els elements del model com els de la realitat, i que ho fan de manera consistent. Així podem usar Prop com a llenguatge de comunicació dins de \mathbf{B} .

Excursió La demostració de les afirmacions anteriors és força tècnica. Cal tenir en compte el paper de les anòmies i de les sinonímies, i cal assegurar que tot valor de $X \in \text{PROP}_{Id}(\mathbf{B})$ realment identifica una realització de \mathbf{B} (i anàlogament per $X \in \text{PROP}_{Id}(\mathbf{A})$).

Identificació i comunicació.

Una *propietat d'identificació especular* d'un *component ben definit*, és útil com a llenguatge de comunicació de les realitzacions del component en qüestió, i de les realitzacions del concepte pertinent.

(57)

4.2 La unió fa la força

El significat de la doble comunicabilitat. L'extensió de les propietats *comunicable* i *estructura comunicable* a tots els components d'un model assegura que el model en qüestió (si els únics elements que cal considerar són els conceptes, els components i les respectives realitzacions) té les propietats estructurals necessàries per a la comunicació perfecta. \rightarrow

Conclusió 41, pàgina 263. (14.3.1. Completesa i solidesa)

Components identificables. Sigui MODEL el model format per la unió de *components ben definits*. Si podem assegurar que cada component té un nom, i que aquest és únic dins el conjunt de components considerat, llavors direm que els components són *identificables* en el MODEL .

Llenguatge de comunicació. Donat un model MODEL format per la unió de *components ben definits*, i tal que aquests components són *identificables*, llavors tenim tots els elements per construir el llenguatge necessari per a la comunicació. Per comunicar una realització n'hi ha prou amb comunicar l'abstracció (que sabem comunicable), i la realització dins d'aquesta (que també és comunicable perquè l'abstracció o bé és un component ben definit, o bé és un concepte modelat amb un component ben definit).

(58)

Model amb components ben definits identificables.

Un model format només amb *components ben definits* i *identificables*²³ és un model comunicable.

Excursió. (Identificació d'ús extern, intern, i doble) L'expansió de l'identificador "trasllada" els noms del model a la realitat, i gràcies a això la comunicació és possible, tot i que des de la realitat els noms "rebutts" poden ser un autèntic galimaties. Per aquest motiu sovint, a part de la propietat d'*identificació especular*, s'exigeix una altra propietat: el nom de cada realització ha de ser comprensible pels observadors d'ambdós mons.

Un cas típic de d'identificació no comprensible pel model són els àlies usats com a argot comunicatiu pels usuaris del sistema. Per exemple, quan enlloc de parlar del client amb tal nom i que viu a tal banda, ens hi referim com "el pesat de cada dilluns". Es tracta d'una *identificació d'ús extern*, usada en l'exterior del sistema, però que no és emprada per a la comunicació.

Un cas típic d'identificació incomprensible pels usuaris del sistema és l'identificador que internament es dona a cada objecte en el moment de crear-lo, sigui com un número d'ordre o com una estampació de temps. Fora del sistema aquesta informació no té cap mena d'utilitat. Es tracta d'una *identificació d'ús intern*, usada internament pel sistema, però que no és emprada per a la comunicació.

La *identificació d'ús extern* i la *identificació d'ús intern* es poden veure com mecanismes individualitzadors que només tenen sentit en un dels mons en joc: en la identificació externa els mecanismes cognitius dels usuaris permeten anar del nom a la realització; en la identificació interna són els mecanismes interns del sistema (visibilitats, índexs, etc) els qui fan la tasca. Per tal que la comunicació sigui efectiva cal un mecanisme d'*identificació d'ús doble* usable en cadascun dels mons.

Si bé res no impedeix usar la *identificació d'ús intern* com a identificació per a la comunicació (simplement cal exigir a l'usuari que s'aprengui o apunti els noms que li proporciona el sistema), el sistema resultant no és gens amigable. D'aquí que és preferible usar per a la comunicació una *identificació d'ús doble* de cara a la comunicació, tot i que potser internament s'usi una *identificació d'ús intern*.

5 Identificació i llenguatge de comunicació

Un model a dos nivells que queda coix. Acabem de veure que un model construït exclusivament amb *components ben definits* és un *model comunicable*; i

²³La majoria de les propietats exigides a un *component ben definit* tracten el component de manera independent a la resta. La propietat de *Comunicable* és l'única propietat dels *components ben definits* que té en compte un context: la del model i la realitat considerada. L'exigència que els *components ben definits* siguin identificables té en compte exactament el mateix context. Així, si bé la propietat de les *Realitzacions comunicables* d'un *component ben definit* demana l'existència d'una propietat d'identificació per a les realitzacions dels components, quan demanem que el *component ben definit* sigui identificable el que estem demanant és que el model resultant de la unió de components ben definits tingui una propietat d'identificació sobre el fragment *MODEL_{Abs}*.

que les propietat d'identificació dels components que el componen formen la base del llenguatge de comunicació que usarem per a la comunicació. Ara ens proposem introduir d'altres elements en el model (i en la realitat): les interrelacions entre aquests components.

5.1 Associacions i visibilitats

5.1.1 Llenguatge dels enllaços

Context de l'anàlisi. Sigui $MODEL'$ el nostre model, construït només amb components ben definits. Siguin B_1 i B_2 dos components del $MODEL'$ tals que $model(A_1) = B_1$ i $model(A_2) = B_2$.²⁴ Sigui una associació $assoc$ entre A_1 i A_2 .²⁵

Model d'una associació. El model d'una associació és una visibilitat directa d'atribut. Per tant $assoc$ la podem modelitzar amb $\forall 1 : B_1 \rightarrow B_2$; amb $\forall 2 : B_2 \rightarrow B_1$; o amb $\forall 1 : B_1 \rightarrow B_2$ i $\forall 2 : B_2 \rightarrow B_1$.

Supòsit d'anàlisi. Suposem que $model(assoc) = \forall 1 : B_1 \rightarrow B_2$. Tot seguit es tracta d'analitzar les propietats de $MODEL = MODEL' \cup \{\forall 1 : B_1 \rightarrow B_2\}$.

Excursió. (Múltiples models d'una associació) Estem considerant el cas que el model d'una associació és una única visibilitat. Això és així perquè ens interessa obtenir un model coherent, i per tant amb modelització unívoca. Malgrat tot sabem que en alguns casos interessa modelitzar una associació binària amb les dues visibilitats possibles. En aquest cas perdem la coherència. Analitzem la situació a l'apartat ??, pàgina ??.

??, pàgina ??

Nom d'un enllaç. La realització d'una associació és un enllaç. El nom d'un enllaç, dins el context de l'associació, és el parell format pels noms dels elements enllaçats.

Nom d'un enllaç dirigit. La realització d'una visibilitat d'atribut és un enllaç dirigit. El nom d'un enllaç dirigit, dins el context de la visibilitat, és el parell format pels noms dels elements enllaçats.

Propietat d'identificació d'una associació. Donada una associació $assoc: A1-A2$, tenim que $PROP_{Id}(assoc) \supseteq PROP_{Id}(A1) \times PROP_{Id}(A2)$.²⁶ Si expressem $assoc$

²⁴Com que els components són ben definits podem assegurar, per la propietat de *comunicable*, que cadascun d'ells és model d'algun concepte A_i , i que aquest concepte només té a B_i com a model.

²⁵En tot el que segueix considerem associacions binàries $M-N$. Més tard ho generalitzarem a associacions $1-N$ i a associacions n -àries.

²⁶Formalment, $PROP_{Id}(X)$ és el conjunt format per tots els conjunts de noms que són propietat d'identificació d' X . Així, l'expressió $PROP_{Id}(assoc) \supseteq PROP_{Id}(A1) \times PROP_{Id}(A2)$ diu que el producte cartesià d'una propietat d'identificació $PROP_{Id}(A1)$ qualsevol sobre $A1$, amb una propietat d'identificació $PROP_{Id}(A2)$ qualsevol sobre $A2$, és una propietat d'identificació $PROP_{Id}(assoc)$ sobre $assoc$. És a dir, $PROP_{Id}(A1) \times PROP_{Id}(A2) \in PROP_{Id}(assoc)$.

$= A1 \times A2$, com es fa sovint, llavors podem escriure $\text{PROP}_{Id}(A1 \times A2) \supseteq \text{PROP}_{Id}(A1) \times \text{PROP}_{Id}(A2)$.²⁷

Esbós de la demostració.

- *Obligatorietat.* Tot enllaç d'assoc interrelaciona una realització del concepte A_1 amb una realització del concepte A_2 . Si A_1 té una propietat d'identificació $\text{Prop}_{Id}(A_1)$ significa que tota realització de A_1 té un nom x (en concret, $(\exists x)x \in \text{Prop}_{Id}(A_1)$); anàlogament per les realitzacions de A_2 . Per tant, per tot enllaç d'assoc els elements enllaçats tenen nom, i conseqüentment podem expressar un enllaç amb el nom de les realitzacions enllaçades: $\{x, y\}: A_1 - A_2$. Així tot enllaç d'assoc té un nom a $\text{Prop}_{Id}(A_1) \times \text{Prop}_{Id}(A_2)$.
- *Unicitat.* Entre dues realitzacions qualssevol $x:A_1$ i $y:A_2$, només hi pot existir un enllaç de l'associació *assoc* (tot i que hi poden existir enllaços per a d'altres associacions). Per tant l'ús del nom dels extrems de l'enllaç com a nom de l'enllaç no pot provocar ambigüitat referencial.

Propietat d'identificació d'una associació en termes dels components. Si tots els components involucrats en una associació són *ben definits* llavors, per la propietat de les *realitzacions identificables*, \rightarrow sabem que tota propietat d'identificació de B també ho és de A . Així, si $\text{model}(A1) = B1$ i $\text{model}(A2) = B2$, tenim que $\text{PROP}_{Id}(A1 \times A2) \supseteq \text{PROP}_{Id}(B1) \times \text{PROP}_{Id}(B2)$. Aquest resultat diu que el nom d'un enllaç és el parell format pels noms dels models dels elements enllaçats.

Propietat d'identificació d'una visibilitat. Donada una visibilitat $v:B1 \rightarrow B2$, tenim que $\text{PROP}_{Id}(v) \supseteq \text{PROP}_{Id}(B1) \times \text{PROP}_{Id}(B2)$.²⁸ Si expressem $v = B1 \times B2$, com es fa sovint, llavors es compleix que:

$$\text{PROP}_{Id}(B1 \times B2) \supseteq \text{PROP}_{Id}(B1) \times \text{PROP}_{Id}(B2).$$
²⁹

Esbós de la demostració.

Els enllaços dirigits que són realització d'una mateixa visibilitat sempre tenen la mateixa orientació. Per tant aquesta no és una propietat de l'enllaç dirigit, sinó que ho és de la visibilitat. El que distingeix un enllaç dirigit d'un altre són els objectes enllaçats. Per tant podem aplicar el mateix raonament que hem usat pels enllaços entre conceptes.

²⁷Aquí l'ordre associat als productes cartesianes no és significatiu, de tal manera que *assoc* es pot expressar tant com $A1 \times A2$ o bé com $A2 \times A1$.

²⁸Formalment, $\text{PROP}_{Id}(X)$ és el conjunt format per tots els conjunts de noms que són propietat d'identificació d' X . Així, l'expressió $\text{PROP}_{Id}(v) \supseteq \text{PROP}_{Id}(B1) \times \text{PROP}_{Id}(B2)$ diu que el producte cartesià d'una propietat d'identificació $\text{Prop}_{Id}(B1)$ qualsevol sobre $B1$, amb una propietat d'identificació $\text{Prop}_{Id}(B2)$ qualsevol sobre $B2$, és una propietat d'identificació $\text{Prop}_{Id}(v)$ sobre v . És a dir, $\text{Prop}_{Id}(B1) \times \text{Prop}_{Id}(B2) \in \text{PROP}_{Id}(v)$.

²⁹Aquí l'ordre associat als productes cartesianes sí és significatiu, de tal manera que $B1 \times B2$ expressa una visibilitat $(B1 \rightarrow B2)$, mentre que $B2 \times B1$ n'expressa una altra $(B2 \rightarrow B1)$.

Comunicació d'enllaços. En considerar tant les associacions com les visibilitats com a productes cartesianes de components ben definits, assegurem la comunicació perfecta dels enllaços (dirigits o no).

Esbós de la demostració.

- Propietats estructurals.
 - *Solidesa.* Els extrems són sòlids; per tant tot enllaç dirigit del model és un possible enllaç de la realitat.
 - *Completesa.* Els extrems són complerts; per tant tot enllaç de la realitat és capturat per un enllaç dirigit del model.
 - *Isomorfia.* Els extrems són coherents; per tant donada una realització d'un dels extrems només hi ha una realització que la modelitza. Així tot enllaç només és modelitzable amb un enllaç dirigit de la visibilitat considerada.³⁰. I tot enllaç dirigit és el model d'un únic enllaç.
- Llenguatge de comunicació.
 - *Sintaxi comuna.* La propietat de les *relitzacions identificables dels components ben definits*, assegura que la *propietat d'identificació* de cada extrem de la *visibilitat* és també la *propietat d'identificació* del concepte que modelitza, que al seu torn és l'extrem de l'*associació* modelitzada per la *visibilitat* considerada. Per tant el producte cartesià de noms en el model dóna el mateix resultat que el producte cartesià de noms en la realitat.
 - *Semàntiques consistents.* Els noms de les realitzacions dels components ben definits compleixen la propietat de les semàntiques consistents. Això assegura que els noms dels enllaços (dirigits o no) com a producte cartesià dels noms dels seus extrems també la compleixen.
 - *Completesa.* Tot enllaç (dirigit o no) té nom, ja que en tenen els elements dels seus extrems.
 - *Referent únic.* El nom d'un enllaç (dirigit o no) és únic, ja que ho és el nom dels seus extrems.

Generalització. Tots els resultats anteriors són fàcilment generalitzables a les associacions n-àries.

³⁰Reocordem que estem considerant que l'associació es modelitza amb una sola visibilitat.

(59)

Llenguatge dels enllaços, dirigits o no (1)

Donada una associació $assoc: A_1 \times A_2$ i una visibilitat que la modelitza, llavors $L_{assoc} = L_{A_1 \times A_2} = Prop_{Id}(A_1) \times Prop_{Id}(A_2)$ és útil com a llenguatge de comunicació dels enllaços, dirigits o no. Aquest resultat es pot generalitzar a les associacions n-àries.³¹

Excursió. (Associacions amb pota a 1) La conclusió anterior es pot matisar: del producte cartesià de les propietats d'identificació dels extrems en podem suprimir una, i només una, sempre i quan la multiplicitat de la pota en qüestió sigui 1.

Semàntica d'una associació. La semàntica d'una associació diu que cada enllaç determina de forma unívoca les realitzacions dels seus extrems. Donada una associació $A-B$ sabem que per cada parell $\langle a : A, b : B \rangle$ hi ha com a molt un enllaç. A més, un enllaç no pot viure sense els seus extrems; és a dir, la determinació d'un enllaç determina de manera unívoca les realitzacions dels conceptes enllaçats. Per tant, tota propietat d'identificació de l'associació és una propietat d'identificació del producte cartesià dels extrems: $Prop_{Id}(A \times B) \subseteq Prop_{Id}(A) \times Prop_{Id}(B)$.

Semàntica d'una visibilitat. Donada una visibilitat $A \rightarrow B$ sabem que per cada parell $\langle a : A, b : B \rangle$ hi ha com a molt un enllaç dirigit. A més, un enllaç dirigit no pot viure sense els seus extrems: l'origen de l'enllaç és qui manté l'enllaç dirigit; l'existència de la destinació de l'enllaç és imprescindible per considerar l'enllaç dirigit com a vàlid.

Llenguatges alternatius. Tot llenguatge usat per a la comunicació dels enllaços (dirigits o no) ha de ser capaç de comunicar no només un enllaç qual-sevol sinó també els extrems enllaçats. En el cas d'usar components ben definits això significa que el nom d'un enllaç ha de poder-se descomposar en el valor d'identificació de cadascun dels seus extrems.

(60)

Llenguatge dels enllaços, dirigits o no (i 2)

Donada una associació $assoc: A_1 \times A_2$ i una visibilitat que la modelitza, tot llenguatge L_{assoc} de comunicació dels enllaços, dirigits o no, s'ha de poder descomposar de la manera següent:

$$L_{assoc} = L_{A_1 \times A_2} = Prop_{Id}(A_1) \times Prop_{Id}(A_2).$$

Aquest resultat es pot generalitzar a les associacions n-àries.³²

³¹Sigui $model(A_i) = B_i$. Recordem que es compleix que per ser els components ben definits, la propietat d'identificació considerada per a B_i també és una propietat d'identificació per a A_i .

³²Sigui $model(A_i) = B_i$. Recordem que es compleix que $Prop_{Id}(A_i) = Prop_{Id}(B_i)$.

5.1.2 Llenguatge de les associacions i visibilitats

Comunicació d'enllaços. La propietat d'identificació d'una *associació* o d'una *visibilitat* és el producte cartesià de les propietats d'identificació dels seus extrems. Això significa que donada una associació o visibilitat, si les considerem com el model d'un conjunt de realitzacions (els enllaços, dirigits o no), llavors tenim un llenguatge per comunicar aquestes realitzacions d'un món a l'altre: n'hi ha prou en comunicar quins elements són els que estan connectats.

La importància del context. El llenguatge de comunicació dels enllaços, $L_{A1 \times A2} = Prop_{Id}(A1) \times Prop_{Id}(A2)$, només té sentit en el context del parell associació/visibilitat considerats. En el moment que la visibilitat s'introdueix com a abstracció dins del MODEL en la comunicació hem d'indicar de quina associació/visibilitat estem parlant. Com a conseqüència el nom d'un enllaç, dirigit o no, és el nom de l'associació o de la visibilitat, més el nom dins d'aquesta abstracció.

Coherència del MODEL. Hem definit MODEL com la unió del MODEL', format només amb components ben definits, amb el model de l'associació considerada. I volem que MODEL sigui coherent. Això exigeix que el nom de l'associació sigui únic en la realitat, que el nom de la visibilitat considerada sigui única en el model, que l'associació tingui un sol model, i que aquest sigui només el model de l'associació en qüestió.

Nom d'una associació. Totes les abstraccions han de tenir nom, i que aquest nom, en el cas de les associacions, no pot ser el producte cartesià dels extrems, ja que entre dos conceptes qualssevol hi poden haver múltiples associacions.

Llenguatge de les abstraccions. La finitud del model $MODEL_{Abs}$ de les abstraccions permet construir el model, coherent, segons els principis de la *Franquícia* i de la *Franquícia obligada*.³³ \rightarrow El resultat és que els noms del model formen el llenguatge de comunicació de les abstraccions.

Conclusió 49, pàgina 274. (14.5. Comunicació i principis de modelització)

5.2 Conceptes associatius

Conceptes associatius i associacions. Un concepte associatiu no és res més que una associació, en la que els enllaços tenen atributs. En termes tècnics un concepte associatiu és una associació predicativa, mentre que les "associacions" són associacions enunciatives. El que canvia, per tant, és la informació que conté, no pas el seu mecanisme individualitzador.³⁴ Així els resultats sobre les associacions també són aplicables als conceptes associatius.

³³Gràcies a què $MODEL_{Abs}$ és finit, podem assegurar fàcilment la completesa i la *Franquícia obligada*, que subsumeix la *Franquícia*

³⁴La diferència és la mateixa que hi ha entre un concepte amb dues propietats o atributs, i un concepte amb tres propietats o atributs.

Identificació d'un concepte associatiu³⁵. Donat un concepte associatiu $A: A_1 - A_2$, tenim que $\text{PROP}_{Id}(A) \supseteq \text{PROP}_{Id}(A_1) \times \text{PROP}_{Id}(A_2)$. Si expressem $A = A_1 \times A_2$, com es fa sovint, llavors podem escriure:

$$\text{PROP}_{Id}(A_1 \times A_2) \supseteq \text{PROP}_{Id}(A_1) \times \text{PROP}_{Id}(A_2).^{36}$$

Conceptes associatius en el model. Un *concepte associatiu* es modelitza amb un component. El que diferencia un component qualsevol X d'un component B que és el model d'un *concepte associatiu* és que les realitzacions de B són també el model d'un enllaç. Això significa que en comunicar una realització de B també hem de comunicar les realitzacions que enllaça. Però si comuniquem aquestes, llavors no cal comunicar res més, perquè entre elles només hi ha un enllaç possible.

Identificació del model d'un concepte associatiu. A partir de les observacions anteriors podem afirmar que donat un concepte associatiu $A: A_1 - A_2$, tal que $\text{model}(A) = B$, $\text{model}(A_1) = B_1$ i $\text{model}(A_2) = B_2$, llavors $\text{PROP}_{Id}(B) \subseteq \text{PROP}_{Id}(B_1) \times \text{PROP}_{Id}(B_2)$. Si expressem $B = B_1 \times B_2$, com es fa sovint, llavors podem escriure:

$$\text{PROP}_{Id}(B_1 \times B_2) \subseteq \text{PROP}_{Id}(B_1) \times \text{PROP}_{Id}(B_2).^{37}$$

Llenguatge de les realitzacions dels conceptes associatius. De nou tenim totes les condicions per aplicar els resultats obtinguts en estudiar les associacions. Tot seguit plasmem aquests resultats.

(61)

Llenguatge de les realitzacions dels conceptes associatius (1)

Donat un concepte associatiu $A: A_1 \times A_2$, i un component B que el modelitza, llavors tenim que el llenguatge $L_A = L_{A_1 \times A_2} = \text{Prop}_{Id}(A_1) \times \text{Prop}_{Id}(A_2)$ és útil com a llenguatge de comunicació dels enllaços, dirigits o no. Aquest resultat es pot generalitzar a les associacions n -àries.³⁸

³⁵ Aquest resultat és l'obtingut amb les associacions, però adaptat als conceptes associatius.

³⁶ Aquí l'ordre associat als productes cartesianes no és significatiu, de tal manera que A es pot expressar tant com $A_1 \times A_2$ o bé com $A_2 \times A_1$.

³⁷ Aquí l'ordre associat als productes cartesianes no és significatiu, de tal manera que B es pot expressar tant com $B_1 \times B_2$ o bé com $B_2 \times B_1$.

³⁸ Sigui $\text{model}(A_i) = B_i$. Recordem que es compleix que per ser els components ben definits, la propietat d'identificació considerada per a B_i també és una propietat d'identificació per a A_i .

Llenguatge de les realitzacions dels conceptes associatius (i 2)

(62)

Donat un concepte associatiu $A: A_1 \times A_2$ i un component B que el modelitza, tot llenguatge L_A de comunicació de les realitzacions del concepte associatiu s'ha de poder descomposar en termes dels llenguatges de les abstraccions pertinents. En concret, $L_A = L_{A_1 \times A_2} = Prop_{Id}(A_1) \times Prop_{Id}(A_2)$. Aquest resultat es pot generalitzar a les associacions n -àries.³⁹

Llenguatge dels conceptes associatius. A l'igual que amb les associacions i visibilitats, entre els mateixos conceptes hi pot haver múltiples conceptes associatius. Per tant cada parell format per un concepte associatiu i el component que el modelitza, ha de tenir el seu propi nom. La comunicació llavors es realitza usant el llenguatge per nivells.

5.3 Especialitzacions

5.3.1 Les realitzacions de les especialitzacions

Notació. Per indicar que un element B és una especialització d'un element A ho expressarem amb $B \subseteq A$.



Model d'una especialització. El model d'una especialització és també una especialització.⁴⁰

Esbós de la demostració.

Sigui una especialització $A_2: A_2 \subseteq A_1$. Llavors, A_2 és un concepte, que té la peculiaritat de ser una especialització de A_1 , la qual cosa significa que tota realització de A_2 també és una realització de A_1 .

Sigui $model(A_2) = B_2$. Per modelitzar la idea d'especialització hem d'assegurar que el model de tota realització de A_2 sigui una realització del model de A_1 . Però donada una realització de A_2 el seu model ha de ser una realització de B_2 , i el model de A_1 és B_1 . Per tant hem, d'assegurar que tota realització de B_2 sigui una realització de B_1 . És a dir $B_2 \subseteq B_1$.

³⁹Sigui $model(A_i) = B_i$. Recordem que es compleix que per ser els components ben definits, la propietat d'identificació considerada per a B_i també és una propietat d'identificació per a A_i .

⁴⁰Aquesta afirmació fins ara la consideràvem com un fet. Aquí el que fem és presentar-la formalment i demostrar la seva correctesa.

Nom de la realització d'una especialització. La realització d'una especialització és una realització compartida per dues abstraccions: ella que especialitza i la que generalitza.⁴¹ La realització és el mateixa, tant si la considerem des de la perspectiva de l'especialització com si la considerem des de la perspectiva de la generalització.⁴² Per tant comparteixen nom.⁴³

(63)

El nom d'una realització vista des de la perspectiva de la *generalització* també és el nom de la realització des de la perspectiva de l'*especialització*.

Noms alternatius. Tota realització d'una *especialització* té com a nom el que té des de la perspectiva de la *generalització*. Però, res no impedeix que en la perspectiva de l'especialització també puguem usar d'altres noms que res tinguin a veure amb la propietat d'identificació de la generalització corresponent. A 15.5.3.3. *Identificació independents* → n'analitzem les possibilitats i conseqüències.

15.5.3.3. Identificació independents, pàgina 308

Exemple 75 (Noms genèrics i noms especialitzats) Els Vehicles poden ser Aquàtics i Terrestres. En la generalització Vehicle ens veiem obligats a especificar al màxim per tal d'evitar confusions. Per exemple, si usem el nom *moto* per referir-nos a les "bicicletes amb motor",⁴⁴ llavors caldrà usar *moto-d'aigua* per referir-nos al vehicle de desplaçament ràpid sobre l'aigua.

Dins l'especialització Aquàtics no té sentit matisar tant. Si usem *moto* està clar que és una *moto aquàtica*. Però si usem *moto-d'aigua* tant li fa en la perspectiva en què ens ho mirem, que no hi ha ambigüitat possible.

Fixem-nos com no només el nom pot canviar en funció de la perspectiva, sinó que també es poden produir col·lisions: un mateix nom, en una perspectiva o en una altra fa referència a elements diferents.

⁴¹Fugim expressament d'emprar els termes *subclasse* i *superclasse* per tractar-se de conceptes purament implementatius.

⁴²Per exemple, si $B \subseteq A$ llavors $a : B$ és tant un objecte $:A$ com un objecte $:B$.

⁴³A 15.5.3.2. *Identificació per restricció de la generalització* → fem el desenvolupament formal d'aquesta observació.

⁴⁴Perdó per l'absurditat de la definició. Però per a l'exemples és suficient.

15.5.3.2. Identificació per restricció de la generalització, pàgina 307

5.3.2 Identificació per restricció de la generalització

Definició. Restricció d'una propietat d'identificació

Direm que una propietat d'identificació $Prop_{Id}(B)$ és una restricció d'una propietat d'identificació $Prop_{Id}(A)$ si i només si :

- **Restricció del domini.** El domini de $Prop_{Id}(A)$ inclou el domini de $Prop_{Id}(B)$. És a dir, $B \subseteq A$.
- **Consistència de valors.** $(\forall b : B) id_{Prop_{Id}(B)}(b) = id_{Prop_{Id}(A)}(b)$

Restricció del domini. La propietat de l'expansió del domini diu que el conjunt de realitzacions de l'element identificat per $Prop_{Id}(B)$ és un subconjunt del conjunt de realitzacions de l'element identificat per $Prop_{Id}(A)$.

Consistència de valors. La propietat de l'expansió de valors diu que la propietat d'identificació $Prop_{Id}(B)$ coincideix arreu amb la propietat d'identificació $Prop_{Id}(A)$. La diferència entre ambdues propietats d'identificació és que la restricció només permet identificar les realitzacions de B (que per la propietat de la restricció de dominis també són realitzacions de A): $Prop_{Id}(A)$ té un domini més extens que $Prop_{Id}(B)$.

Notació. Usarem $Prop_{Id}(B) \subseteq Prop_{Id}(A)$ per expressar que $Prop_{Id}(B)$ és una restricció de $Prop_{Id}(A)$.



Restricció sobre un domini. Donada una propietat d'identificació $Prop_{Id}(A)$, anomenarem *restricció de $Prop_{Id}(A)$ sobre el domini de B* a tota propietat d'identificació $Prop_{Id}(B)$ tal que $Prop_{Id}(B) \subseteq Prop_{Id}(A)$.

Excursió. (Propietat d'identificació com a relació) Una propietat d'identificació es pot veure com una relació entre el conjunt de les realitzacions de l'element que identifica i el conjunt dels noms. Així, tota propietat d'identificació es pot expressar com el conjunt de parells $\langle \text{realització}, \text{nom} \rangle$. Formalment, $Prop_{Id}(A) = \{ \langle a, \text{nom} \rangle \mid a \in A \}$.

En veure les propietats d'identificació com a conjunts, i els elements com a conjunts de realitzacions, podem aplicar les operacions de conjunts sobre elles. Per exemple, una restricció $Prop_{Id}(B) \subseteq Prop_{Id}(A)$ es pot expressar com: $(\exists X)(A = B \cup X) \wedge (Prop_{Id}(A) = Prop_{Id}(B) \cup Prop_{Id}(X))$.

Des d'aquesta perspectiva, una restricció d'una propietat d'identificació és la projecció d'aquesta propietat d'identificació sobre el domini de la restricció.

Notació. Recordem que $X \in Prop_{Id}(B)$ significa que X és UNA propietat d'identificació de B . Però això no treu que B pugui tenir d'altres



propietats d'identificació.

Identificació d'una especialització. Donada una propietat d'identificació qualsevol sobre una generalització, la *restricció* al domini de l'especialització és una propietat d'identificació d'aquesta especialització.⁴⁵

$$A_1 \subseteq A_2 \wedge Prop_{Id}(A_1) \subseteq Prop_{Id}(A_2) \implies Prop_{Id}(A_2) \subseteq Prop_{Id}(A_2)$$

Esbós de la demostració.

Sigui $A_1 \subseteq A_2$, i sigui una propietat d'identificació $Prop_{Id}(A_1)$ sobre la generalització. Per cada realització de A_1 la propietat d'identificació $Prop_{Id}(A_1)$ assigna, com a valor d'identificació, un conjunt de noms (*obligatorietat* de la propietat d'identificació); a més no hi ha cap nom que aparegui en més d'un conjunt de noms (*unicitat* de la propietat d'identificació).

Per tant, en concret, assigna un conjunt de noms a cada realització de A_2 , i els noms no es comparteixen entre realitzacions. Per tant tota restricció de $Prop_{Id}(A_1)$ sobre A_2 és una propietat d'identificació sobre A_2 .

5.3.3 Identificació independents

Ús de propietats d'identificació independents. Si bé l'ús de propietats d'identificació per a una especialització que siguin independents de la propietat d'identificació de la generalització pertinent és possible, \rightarrow no sembla que aporti cap avantatge significatiu.

Exemple 75. Noms genèrics i noms especialitzats,
pàgina 306

? (43)

Quins avantatges i inconvenients té usar una propietat d'identificació per a una especialització independent de la propietat d'identificació de la generalització pertinent?

15.2. Llenguatge per nivells, pàgina 282

Llenguatge per nivells. Per analitzar la resposta a aquesta pregunta cal recordar que com que estem usant el llenguatge per nivells. \rightarrow

El cas econòmic. Sigui una especialització $B_2 \subseteq B_1$ i sigui una $Prop_{Id}(B_2)$ tal que $Prop_{Id}(B_2) \subseteq Prop_{Id}(B_1)$. Llavors, donada una realització $b : B_2$ només hem de considerar, a part dels noms de les abstraccions, un sol nom per a la realització.

Demostració.

- Considerem els següents noms:

Pàgina 306

⁴⁵ Aquest resultat formalitza la idea intuïtiva, presentada més amunt, \rightarrow que el nom de la generalització és també el nom de l'especialització.

- x . El nom de B_1
- y . El nom de B_2
- z . El nom dins de B_2
- Per a comunicar la realització com un element de l'especialització cal emprar $\langle y, z \rangle$.⁴⁶
- Per a comunicar la realització com un element de la generalització cal emprar $\langle z, z \rangle$.
- En aquest cas totes les combinacions tenen significat.
- En resum, *donada una realització tenim un sol nom, compartit per totes les perspectives*. En la comunicació indiquem quina perspectiva cal considerar.

Pèrdua d'economia. Sigui una especialització $B_2 \subseteq B_1$ i $Prop_{Id}(B_2)$ una propietat d'identificació independent de $Prop_{Id}(B_1)$. Llavors, donada una realització $b : B_2$ hem de considerar, a part dels noms de les abstraccions, dos noms diferents per a la realització.

Demostració.

- Considerem els següents noms:
 - x . El nom de B_1
 - y . El nom de B_2
 - z . El nom dins de B_1
 - t . El nom dins de B_2
- Per a comunicar la realització com un element de l'especialització cal emprar $\langle y, z \rangle$.
- Per a comunicar la realització com un element de la generalització cal emprar $\langle x, t \rangle$.
- La combinació $\langle x, t \rangle$ no té cap mena de sentit. Què passa però amb la combinació $\langle y, z \rangle$?
- En resum, *donada una realització tenim un nom diferent per a cada una de les perspectives*. En la comunicació indiquem quina perspectiva cal considerar.

La independència no evita la dependència. És important observar com davant d'una especialització $B_2 \subseteq B_1$ on $Prop_{Id}(B_2)$ és independent de $Prop_{Id}(B_1)$, continuem tenint que $Prop_{Id}(B_1)$, és una propietat d'identificació de B_2 . Per tant, segons la nomenclatura del paràgraf anterior, la combinació $\langle y, t \rangle$ és perfectament vàlida i és sinònim de $\langle x, z \rangle$.

⁴⁶Recordem que en el llenguatge per nivells comuniquem per separat el nom de l'abstracció, i el nom de la realització. En aquest cas l'abstracció és y i la realització dins d'aquesta abstracció és z .

Factorització de la identificació. La propietat d'identificació de la generalització sempre la necessitem; i sabem que també és propietat d'identificació de totes les seves especialitzacions. És a dir, la propietat d'identificació de les especialitzacions *factoritza* cap a la generalització.

Excursió. (Factorització de la identificació) Suposem el principi de la *Identificació atributiva*.⁴⁷ Aquest principi que veurem més endavant diu que les propietats d'identificació són atributs dels components. Així, en una generalització ha d'aparèixer l'atribut identificatiu, que es "propaga"⁴⁸ a totes les seves especialitzacions.

Conclusions sobre la independència. Podem tenir una especialització amb una propietat d'identificació independent de la propietat d'identificació de la generalització pertinent? Sí. Però tot i així necessitem també la propietat d'identificació dependent: com a identificació de l'especialització sempre hem de considerar la identificació de la generalització.

5.3.4 Comunicació de les especialitzacions

Context de l'anàlisi. Sigui MODEL' el nostre model, construït només amb components ben definits. Sigui B_1 un component del MODEL' tal que $\text{model}(A_1) = B_1$.⁴⁹ Suposem que afegim un *concepte especialitzat* $A_2 \subseteq A_1$.

Objectiu que perseguim. El que pretenem és introduir a MODEL' el model de l'especialització A_2 , de tal manera que el model resultant continuï sent model a dos nivells, i que continuem tenint un llenguatge per nivells vàlid per a la comunicació.

Model amb una especialització. Tot seguit es tracta d'analitzar les propietats de $\text{MODEL} = \text{MODEL}' \cup \{B_2: B_2 \subseteq B_1\}$, sota el supòsit que B_2 sigui un *component ben definit*.

Identificació de l'especialització introduïda a MODEL' . El nostre model MODEL' està format només per components ben definits, i per tant tots tenen identificació especular. Per tant, en introduir l'especialització B_2 podem prendre la propietat d'identificació especular de la generalització per a identificar l'especialització.⁵⁰

⁴⁷Vegeu l'apartat 19.5. Identificació atributiva, pàgina 379.

⁴⁸No usem l'expressió "hereda" per no confondre'ns amb els mecanismes d'herència dels llenguatges OOP.

⁴⁹Com que els components són ben definits podem assegurar, per la propietat de *comunicable*, que cadascun d'ells és model d'algun concepte A_i , i que aquest concepte només té a B_i com a model.

⁵⁰formalment el que prenem és la seva restricció al domini de l'especialització.

En introduir una especialització només ens hem de preocupar que el nou component sigui *comunicable* i tingui *estructura comunicable*.⁵¹, →

Si prenem com a propietat d'identificació la propietat d'identificació de la generalització corresponent obtenim les propietats de *realitzacions identificables* i *identificació per expansió*. El resultat és que el component introduït és un component ben definit.

(64)

Definició pàgina 296. (15.4. Component ben definit)

Construcció d'una especialització. Si construïm les especialitzacions com a subconjunts de components presents en el model, llavors la propietat d'*Estructura comunicable* està assegurada.

Introducció d'una especialització.

Per construir una especialització cal:

- *Assegurar la comunicabilitat.* Cal donar un nom únic a l'especialització dins del MODEL considerat.
- *Estructura de subconjunt.* Cal definir l'especialització com el subconjunt d'un component ben definit present a MODEL.
- *Identificació heretada.* Com a propietat d'identificació de l'especialització prenem la propietat d'identificació de la generalització.

(65)

Assegurar la comunicabilitat. Donat un component qualsevol, s'en poden definir moltes especialitzacions diferents. D'aquí que calgui un nom diferent per cadascuna d'elles.

Estructura de subconjunt. Com hem vist aquesta és l'única propietat estructural que ens cal assegurar per tal de construir l'especialització com a component ben definit.

Identificació heretada. Si per a la comunicació de les realitzacions d'una especialització usem el mateix llenguatge que la seva generalització, com que sabem que aquest segon és un llenguatge vàlid per a la comunicació, llavors tenim tots els problemes comunicatius resolts.

⁵¹ Les propietats de *comunicable* i d'*estructura comunicable* formen part de la definició de *component coherent*, pàgina 296.

(66)

Llenguatge de les realitzacions d'una especialització.

Per a la comunicació de les realitzacions d'una especialització A_2 : $A_2 \subseteq A_1$ podem usar el mateix llenguatge que en la generalització: $L_{A_2} = L_{A_1} = Prop_{Id}(A_1)$.⁵²

Excursió. (Especialització que és un concepte associatiu) Sigui una especialització $B \subseteq A$ i $C \subseteq A$. Suposem que C és el model d'un concepte associatiu entre els conceptes X i Y , que al seu torn es modelitzen amb els components X i Y . Sigui una realització $c : C$.

La factorització de la identificació significa que $Prop_{Id}(C) = Prop_{Id}(A)$.⁵³ Per tant $c : C$ queda identificat amb el valor d'identificació que li dona la propietat d'identificació $Prop_{Id}(A)$ de la generalització.

Però per tractar-se C del model d'un concepte associatiu, sabem que tota propietat d'identificació de C s'ha de poder expressar com $Prop_{Id}(X) \times Prop_{Id}(Y)$.

Les condicions exigides a la propietat d'identificació en el cas de l'especialització i en el cas del concepte associatiu no són reconciliables. Per una banda demanem que $c : C$ es pugui identificar simplement per ser un cas particular de A . Per l'altra, demanem que s'identifiquin dos objectes $x : X$ i $y : Y$. En el primer cas $c : C$ és independent de $x : X$ i $y : Y$; en el segon cas $c : C$ no pot existir si *prèviament* no existeixen $x : X$ i $y : Y$.

En conclusió el model presentat és incorrecte. El model correcte és el de partida, però enlloc de tenir $C \subseteq A$ tenim $A \subseteq Z$, on Z és un nou component "lligat" a C .⁵⁴

Excursió. (Especialització d'un concepte associatiu) Sigui C un component que modelitza un concepte associatiu entre A i B , que al seu torn es modelitzen amb els components A i B . Sigui X una especialització de C . sigui una realització $x : X$.

Per una banda tenim que tota propietat d'identificació de C s'ha d'expressar com $Prop_{Id}(A) \times Prop_{Id}(B)$. Per l'altra, tota propietat d'identificació de C també ho és de X . Així tenim que $Prop_{Id}(X) = Prop_{Id}(A) \times Prop_{Id}(B)$.

Si per a X considerem una propietat d'identificació independent de la propietat d'identificació de C , llavors podríem comunicar $x : X$ sense coneixement dels objectes $a : A$ i $b : B$ que enllaça, la qual cosa es contradiu amb les propietats de C com a concepte associatiu

Per tant concloem que el model és vàlid sempre que l'especialització no tingui cap propietat d'identificació independent de la seva generalització.

⁵²Sigui $model(A_i) = B_i$. Recordem que es compleix que $Prop_{Id}(A_i) = Prop_{Id}(B_i)$.

⁵³Recordem la notació. El que ens diu aquesta expressió és que tota propietat d'identificació de A serveix com a propietat d'identificació de C .

⁵⁴Si $model(Z) = Z$ i $model(C) = C$, el que estem dient és que el concepte Z està associat al concepte C . Una altra cosa és amb quina visibilitat modelitzem aquesta associació.

Excursió. Models forçats per la identificació

- Les especialitzacions no poden ser conceptes associatius
- Podem especialitzar un concepte associatiu sempre que mantinguem com a propietat d'identificació de l'especialització la propietat d'identificació de la generalització

(67)

6 La comunicació és possible

Conclusions planeres. Tot seguit plantegem les conclusions d'aquest capítol en termes planers. Entre parèntesis fem referència al concepte tècnic associat. Cal tenir present que hi ha condicions que cal exigir des d'un punt de vista tècnic, que no apareixen en l'exposició planera presentada.

Suposem que hem fet bé les coses. Considerem que els components, com a model, capturen totes les realitzacions del concepte modelitzat, i tenen l'estructura necessària per a la comunicació. (Propietats estructurals dels components ben definits).

Mecanisme d'individualització. Per tal de poder comunicar les realitzacions d'un component exigim que sobre aquest component tinguem un mecanisme per anomenar de manera no ambigua cadascuna de les seves realitzacions. (Propietat d'identificació).

El nom de les realitzacions és útil. El nom usat per expressar un objecte realització d'un component ha de servir també per expressar l'element de la realitat (és a dir, la realització d'un concepte) modelitzada amb l'objecte en qüestió. (Propietat d'identificació especular).

Què és un identificador. En termes planers, l'*identificador* d'una realització és el seu nom, únic en el conjunt de realitzacions de la mateixa abstracció. I exigim que tota realització tingui un nom amb aquestes característiques. (Propietat d'identificació).

El paper dels identificadors. Els *identificadors* sobre els diferents components tenen totes les propietats exigides, i per tant esdevenen el mecanisme bàsic de la comunicació. (Components ben definits).

Interrelacions entre els components. Les associacions, visibilitats, conceptes associatius i especialitzacions es poden comunicar a partir dels mecanismes de comunicació dels components.

Identificadors i models. La importància dels *identificadors* és gran. Els usem com a mecanisme de la comunicació, però la seva presència impossibilita determinats models, o els hi posa condicions restrictives.

Conclusió. Els identificadors resolen el problema de la comunicació entre mons diferents.

Principis i definicions del capítol

Definicions

Component ben definit, 296

Espai d'identificació, 290

Identificador d'una realització, 290

Model d'un conjunt d'entitats, 294

Propietat d'identificació, 289

Propietat d'identificació especular, 294

Propietat d'identificació textual, 291

Restricció d'una propietat d'identificació, 307

Part IV

Construim ponts

Part IV: Taula de continguts

16 Identificació robusta	321
1 Recuperació de realitzacions	323
2 Identificació interna i externa	326
3 Els identificadors en els ES	327
4 Contracte robust	329
Principis	331
Definicions	331
17 Flux comunicatiu	333
1 Arquitectura de tres capes	337
2 Flux comunicatiu extern	340
3 El flux comunicatiu intern	348
4 Encaixada	352
Principis	356
Definicions	356
18 ^{GLS} Identificadors i nova Inscripció (caminada)	357
1 La caminada com a identificador	357
2 La caminada com a objecte	359
19 Àmbits d'identificació	363
1 Identificació i espai d'identificació	366
2 Els identificadors en el MC	367
3 Els identificadors en el MComp	374
4 Identificadors i repositoris	375
5 Identificació atributiva	379
6 Visibilitat dels àmbits d'identificació	381
Principis	383
Definicions	383

20	Generació d'identificadors	385
1	Mecanismes de generació	391
2	Introducció a la GPC	392
3	GPC amb un sol repositori	399
4	GPC amb múltiples repositoris	409
5	Generació per comprovació: conclusions	413
6	Generació única	413
7	Principi de la <i>Individualització efectiva</i>	416
	Principis	418
	Definicions	418
21	Identificació: conclusions	419
1	Identificadors: conclusions	421
2	^{GLS} La identificació d'Inscripcions	424
22	^{GLS} Contracte de l'ES <i>nouParticipant</i> (nom, edat)	425
1	Contracte de partida	428
2	Àmbits d'identificació	429
3	Desenvolupament àgil	436
4	Ús de la <i>Presentació</i>	437
5	Conclusions sobre el contracte	439
6	Retorn a l'especificació	440

Capítol 16

Identificació robusta

1	Recuperació de realitzacions	323
1.1	Recuperació exhaustiva	323
1.2	Recuperació referencial	324
1.3	Recuperació i emmagatzematge	325
2	Identificació interna i externa	326
3	Els identificadors en els ES	327
3.1	Semàntica de l'identificador	327
3.2	Robustesa dels identificadors	327
3.3	Identificació robusta	329
4	Contracte robust	329
	Principis	331
	Definicions	331

Contingut detallat del capítol 16

1 Recuperació de realitzacions	323
1.1 Recuperació exhaustiva	323
Visibilitat del mecanisme de recuperació	323
Apunts sobre la notació	323
1.2 Recuperació referencial	324
Recuperació referencial i recuperació exhaustiva	324
El perquè del nom	324
Apunts sobre la notació	324
1.3 Recuperació i emmagatzematge	325
1.3.1 La recuperació és selecció	325
Emmagatzematge, visibilitat i recuperació	325
1.3.2 Recuperació en els multiobjectes	325
Selecció en els multiobjectes	325
El <code>find()</code> com a mecanisme de recuperació referencial	325
1.3.3 Recuperació referencial en els repositoris	325
Repositoris centralitzats	325
Repositoris distribuïts	325
2 Identificació interna i externa	326
Excursió. (Tipus d'identificació i ús d'identificació)	326
Responsabilitats en la identificació externa	326
Responsabilitats en la identificació interna	326
3 Els identificadors en els ES	327
3.1 Semàntica de l'identificador	327
Semàntica d'un identificador, per l'usuari	327
Restricció de la petició de creació de referència	327
Semàntica d'un identificador, pel sistema	327
3.2 Robustesa dels identificadors	327
Petició correcta d'ús de referència	327
Robustesa del sistema	327
Tenir i oferir	327
Petició de creació de referències amb identificació externa	328
Excursió. (Estructura, existència i recuperació)	328
3.3 Identificació robusta	329
Necessitat de la comprovació d'identificador	329
Identificació robusta	329
Usabilitat de la identificació	329
Responsabilitat del client	329
4 Contracte robust	329
La identificació robusta coma a cas particular	329
Contracte robust	330
Contracte robust	330
Principis	331
Definicions	331

1 Recuperació de realitzacions

1.1 Recuperació exhaustiva

Definició. Recuperació exhaustiva

Direm que un component té recuperació exhaustiva si el sistema ofereix els mecanismes suficients per poder obtenir (recuperar) potencialment qualsevol de les realitzacions d'aquest component.

Exemple 76 (Recuperació exhaustiva) Sigui C un component que volem que tingui recuperació exhaustiva. Qualsevol d'aquests mecanismes l'assegura, tot i que n'hi pot haver-ne d'altres igualment vàlids.

- **Iterador.** Mecanisme que proporciona un iterador sobre totes les realitzacions de C
- **Recuperació global.** Es pot obtenir una estructura única (similar al `list` d'alguns llenguatges de programació) que conté totes les realitzacions de C ¹
- **Estadístic.** Recuperació individual aleatòria d'una realització de C . Cal assegurar, però, que l'estadístic que controla l'aleatorietat és fiable, en el sentit que tota realització de C serà finalment recuperada (potser després de múltiples peticions de recuperació)
- Etc

Visibilitat del mecanisme de recuperació. La definició de recuperació exhaustiva sobre un component exigeix l'existència d'un mecanisme que permeti recuperar potencialment tota realització d'aquest component. El mecanisme concret, però, pot ser totalment intern al sistema (i per tant *ocult* a l'exterior), o pot ser *publicat* en la seva interfície.

Notació. Direm que un objecte $a:A$ té recuperació exhaustiva si la té el component A del qual n'és realització.



Apunts sobre la notació. La recuperació exhaustiva s'ha definit com una propietat sobre un component. La notació presentada diu que, com a conseqüència, les realitzacions del component també mantenen aquesta propietat. De fet es tracta d'un clar abús del llenguatge: no té sentit aplicar una propietat que afecta un col·lectiu sobre un sol element. Malgrat tot aquest abús terminològic no ha de portar confusió i és consistent amb la doble visió objecte-component, realització-abstracció, unitat-col·lectiu.

¹Sovint serà convenient que el que contingui aquesta estructura siguin còpies o rèpliques de les realitzacions de C .

1.2 Recuperació referencial

Definició. Recuperació referencial

Direm que un component té recuperació referencial si el sistema ofereix els mecanismes suficients per poder obtenir (recuperar) potencialment qualsevol de les realitzacions d'aquest component a partir del seu identificador.

Recuperació referencial i recuperació exhaustiva. La *recuperació referencial* és un cas particular de la *recuperació exhaustiva*: la particularitat radica en què la recuperació es fa a partir de l'identificador.

El perquè del nom. Si un component té *recuperació referencial* llavors l'identificador d'una realització n'és la seva referència. D'aquí el nom de la propietat.

Exemple 77 (Recuperació referencial) Per tal que el sistema tingui recuperació referencial del component *Caminada* cal que algun component ofereixi una operació similar a :

`obtenirCaminada(id: Id-Caminada) : <Caminada>2`

tal que donat un valor *id* de la propietat d'identificació *Id-Caminada* retorni la *c:Caminada* única tal que el seu valor per la propietat d'identificació[→] és justament *id*.³

Si abusem del llenguatge podem dir que `obtenirCaminada(...)` és una operació que donat un identificador de *Caminada* retorna la *c:Caminada* identificada amb aquest identificador.

15.3. Propietat d'identificació, pàgina 288



Notació. Direm que un objecte *a:A* té *recuperació referencial* si la té el component *A* del qual n'és realització.



Notació. Direm que una *propietat d'identificació* sobre el component *A* té *recuperació referencial* si la té el component *A*.

Apunts sobre la notació. La *recuperació exhaustiva* s'ha definit com una propietat sobre un component. La notació presentada diu que, com a conseqüència, les realitzacions del component també mantenen aquesta propietat.

²*Id-Caminada* no és cap component, i per tant *id* no és cap objecte. De fet *id* és un literal que permet distingir un objecte *c:Caminada* dels altres objectes *:Caminada*. Hem fet doncs un abús del llenguatge per emfasitzar que *id* és un valor per a la propietat d'identificació, i no pas un literal qualsevol.

³Per aquell lector que no es vulgui submergir en les terres pantanoses[→] podem definir la *propietat d'identificació* com aquella propietat que en donar-li un valor diferència o individualitza (identifica) una realització de les altres realitzacions de la mateixa abstracció.

15. Identificadors, pàgina 277

De fet es tracta d'un clar abús del llenguatge: no té sentit aplicar una propietat que afecta un col·lectiu sobre un sol element. Malgrat tot aquest abús terminològic no ha de portar confusió i és consistent amb la doble visió objecte-component, realització-abstracció, unitat-col·lectiu. Per altra banda, en ser l'identificador una referència a l'objecte, podem fer un nou abús i subsumir en la referència totes les propietats de l'objecte.

1.3 Recuperació i emmagatzematge

1.3.1 La recuperació és selecció

Emmagatzematge, visibilitat i recuperació. L'emmagatzematge de realitzacions d'un component C es correspon amb l'existència d'una *visibilitat* (d'atribut) entre l'emmagatzemador i les realitzacions emmagatzemades. Per tant la recuperació de realitzacions emmagatzemades dins del sistema significa accedir a les realitzacions recuperades a través de la *visibilitat d'emmagatzematge*.

Recuperar una realització significa enviar un missatge de *selecció* sobre la *visibilitat d'emmagatzematge* corresponent

(68)

1.3.2 Recuperació en els multiobjectes

Selecció en els multiobjectes. Els multiobjectes ofereixen una operació de selecció: el `find()`.[→] Tal com s'ha presentat aquesta operació correspon a un mecanisme de *recuperació referencial*.

5.Multiobjectes, pàgina 87

El `find()` com a mecanisme de recuperació referencial. Més concretament, sigui una visibilitat $A \rightarrow B^*$; i sigui id l'identificador de B . Llavors assumim que el multiobjecte que modelitza la multiplicitat de la visibilitat $f : A \rightarrow B^*$ ofereix una operació `find` tal que donat l'identificador d'una realització de B si aquesta realització existeix dins de M la retorna (altrament no retorna res). Formalment, si enviem el missatge $find_B$ a través de l'enllaç dirigit $f : A \rightarrow B^*$:

$$find_B : id_B \rightarrow B$$
$$find_B(id) = b \Leftrightarrow b.id = id \wedge b \in \{x | f(a, x)\}$$

1.3.3 Recuperació referencial en els repositoris

Repositoris centralitzats. La recuperació referencial en el cas de *repositoris centralitzats* [→] es resol amb el mètode `find()` del repositori.

Definició pàgina 148. (9.4. Repositoris)

Repositoris distribuïts. En el cas de *repositoris distribuïts* [→] cal primer cercar el (fragment de) repositori on suposadament hi ha l'objecte que es vol recuperar. Si no tenim manera de conèixer *a priori* quin és aquest (fragment de) repo-

Definició pàgina 148. (9.4. Repositoris)

sitori, no queda més remei que intentar la recuperació en *tots*⁴ els repositoris del component en qüestió.

2 Identificació interna i externa

Definició. Identificació interna

Quan en la creació d'una nova realització d'un component C l'identificador del nou objecte sempre és generat pel sistema, direm que C és d'identificació interna.

Definició. Identificació externa

Quan en la creació d'una nova realització d'un component C l'identificador del nou objecte sempre és proporcionat per l'usuari extern del sistema, direm que C és d'identificació externa.

Definició. Identificació mixta

Quan en la creació d'una nova realització d'un component C alguns cops es proporciona externament l'identificador i en d'altres es genera internament, direm que C és d'identificació mixta.

La *identificació mixta* es pot veure com la coincidència sobre un mateix component C de dos sistemes d'identificació: l'extern i l'intern. Per tant, l'anàlisi que segueix es limita a estudiar els casos intern i extern.

Excursió. (Tipus d'identificació i ús d'identificació) En l'excursió 4.2.Excursió. (Identificació d'ús extern, intern, i doble), pàgina 298,[→] s'ha introduït la identificació d'ús intern, extern o doble. La distinció rau en qui és l'usuari de la propietat d'identificació.

La identificació interna, externa o mixta introduïda ara es defineix en funció de qui crea o defineix el valor d'identificació per a cada realització.

Segurament el més natural és que, per exemple, una identificació interna sigui també d'ús intern, però res no impedeix fer combinacions més o menys estrambòtiques.

Responsabilitats en la identificació externa. En cas d'*identificació externa* correspon a l'usuari del sistema la generació dels identificadors; el sistema no se n'ha de preocupar pas.

Responsabilitats en la identificació interna. En cas d'*identificació interna* correspon al sistema la generació dels identificadors.

⁴No cal en tots els repositoris; és suficient en tots els fragments del repositori distribuït.

3 Els identificadors en els ES

3.1 Semàntica de l'identificador

Semàntica d'un identificador, per l'usuari. L'aparició d'un identificador en un ES només pot ser per dos motius: l'usuari indica al sistema que usi com a argument l'objecte identificat amb el valor donat, o bé l'usuari demana al sistema que creï un objecte que per afutures referències expressarem amb el valor d'identificació donat. En el primer cas direm que l'usuari fa una *petició d'ús d'una referència*; en el segon cas que l'usuari fa una *petició de creació una referència*.

Restricció de la petició de creació de referència. La *petició de creació d'una referència* només és possible sota el context d'*identificació externa*.

Semàntica d'un identificador, pel sistema. Davant *petició d'ús d'una referència* el sistema ha de fer la *recuperació* de l'objecte indicat amb l'identificador. Davant d'una *petició de creació d'una referència* el sistema respon amb la *creació* d'un nou objecte a partir de la informació donada.

3.2 Robustesa dels identificadors

Petició correcta d'ús de referència. Sigui un ES que fa una petició d'ús d'una referència. Pel sistema aquesta petició desemboca en la recuperació d'un objecte. Per tant, en el contracte de l'ES hi ha d'haver una PRE que exigeixi que la recuperació sigui realment possible; és a dir, el contracte ha d'exigir que l'identificador donat és un valor correcte.

Robustesa del sistema. Tot sistema robust ha d'oferir els mecanismes necessaris per tal que els usuaris puguin assegurar i/o comprovar les PRE dels diferents ES.⁵ D'aquí la necessitat que el sistema ofereixi un mecanisme de *comprovació d'identificador* per cada identificador que aparegui en algun ES.

Definició. Mecanisme de comprovació d'identificador

El mecanisme de comprovació d'identificador és un mecanisme ofert pel sistema que donat un identificador ens diu si la recuperació referencial sobre el valor d'identificació donat és possible o no.

Tenir i oferir. El mecanisme de *comprovació d'identificador* no és un mecanisme intern del sistema; és un mecanisme extern: d'aquí que en la definició es parla d'un mecanisme *ofert* i no pas d'un mecanisme *present*. Per tant, el mecanisme de *comprovació d'identificador* ha d'aparèixer en forma d'ES.⁶

⁵Aquesta és la idea del *Desenvolupament sota contracte*:[→] els clients són responsables de comprovar les PRE; i per tant han de tenir els mecanismes per poder fer aquestes comprovacions.

⁶Generalment serà un únic ES; però la definició no ho exigeix.

Exemple 78 (Petició d'ús de referència)

novaInscripció(nom-Caminada) és un ES que demana crear una nova Inscripció, amb identificació interna, associada a la $c:Caminada$ identificada amb $nom-Caminada$. Per tant caldrà una PRE com la següent:

$$(\exists c:Caminada)^7(c.nom = nom-Caminada)$$

Per donar suport a aquesta PRE, en el disseny oferim el mètode existent($nom-Caminada$) que retorna cert quan en el sistema⁸ hi ha una $c:Caminada$ amb $c.nom=nom-Caminada$.

En aquest cas el mecanisme de comprovació d'identificador és un únic mètode accessible des de l'exterior⁹ que es limita a comprovar l'existència d'una realització amb el nom donat.

Petició de creació de referències amb identificació externa. En el cas d'una petició de creació de referències usant el mecanisme d'identificació externa, cal que l'ES que fa la petició tingui una PRE que exigeixi que l'identificador donat sigui correcte i únic dins el sistema.¹⁰ Per tant en el cas d'una petició de creació de referències també cal que el sistema ofereixi el mecanisme de comprovació d'identificador.

Exemple 79 (Petició de creació de referència)

novaCaminada(nom-Caminada), on $nom-Caminada$ és l'identificador de la nova $c1:Caminada$, exigeix una PRE que asseguri que en el sistema no hi ha cap altra $c2:Caminada$ identificada amb $nom-Caminada$:

$$(\nexists c:Caminada)^{11}(c.nom = nom-Caminada)$$

Per donar suport a aquesta PRE, en el disseny oferim el mètode existent($nom-Caminada$) que retorna cert quan en el sistema¹² hi ha una $c:Caminada$ amb $c.nom=nom-Caminada$.

Excursió. (Estructura, existència i recuperació) La comprovació d'identificador engloba dues tasques ben diferenciades: la comprovació d'estructura (que comprova que el valor estigui "ben format") i la comprovació d'existència. Aquí considerem que sempre fem ambdues tasques a l'ensams,

⁷Aquest existencial caldria restringir-los a l'univers de quantificació pertinent, que en aquest cas és l'àmbit d'identificació, un concepte introduït en la pàgina 376. →

⁸Pròpiament en l'àmbit d'identificació.

⁹O bé existent($nom-Caminada$) és un ES, o bé tenim un ES que es limita a cridar-lo.

¹⁰Pròpiament, dins l'àmbit d'identificació.

¹¹Aquest existencial caldria restringir-los a l'univers de quantificació pertinent, que en aquest cas és l'àmbit d'identificació, un concepte introduït en la pàgina 376. →

¹²Pròpiament en l'àmbit d'identificació.

però res impedeix fer un plantejament on ambdues comprovacions es facin en dues tasques independents.

Per altra banda, sovint la *comprovació d'existència* va seguida de la *recuperació*. A més, en molts casos el mecanisme per realitzar la *comprovació d'existència* es basa en la *recuperació*: el responsable $r:\mathbb{R}$ de fer la *comprovació d'existència*, en rebre la petició d'aquesta comprovació, intenta fer la *recuperació*; si la recuperació ha tingut èxit retorna *cert*; altrament retorna *fals*.

En el cas exposat en el que la *comprovació d'existència* oculta una *recuperació*, podem donar la responsabilitat de la *recuperació* al responsable $r:\mathbb{R}$ de la *comprovació d'existència*. Per fer-ho cal demanar a $r:\mathbb{R}$ que quan comprovi l'existència, en cas de tenir èxit emmagatzemi internament la realització trobada. D'aquesta manera quan $r:\mathbb{R}$ rep una petició de recuperació, abans de res mira si el darrer objecte recuperat (i emmagatzemat internament) és el desitjat. Si ho és ja ho tenim, sinó s'encarrega de fer la recuperació.

3.3 Identificació robusta

Necessitat de la comprovació d'identificador. Sigui com sigui, tant si la identificació és externa com interna; tant si usem o creem una referència,¹³ l'aparició d'un identificador com a argument d'un ES exigeix dotar al sistema d'un mecanisme de *comprovació d'identificació*.

Identificació robusta. El sistema ha d'assegurar que, per tot identificador que aparegui en un ES, hi ha un mecanisme de *comprovació d'identificador* visible des de la interfície del sistema.

Principi



Usabilitat de la identificació. El principi de la *Identificació robusta* diu que els únics sistemes d'identificació útils¹⁴ de cara a la comunicació entre el sistema i l'exterior, són aquells que permeten fer les comprovacions necessàries sobre les PRE dels ES.

Responsabilitat del client. El principi de la *Identificació robusta* exigeix que els mecanismes de comprovació apareguin a la interfície del sistema. Això és coherent amb el desenvolupament sota contracte,[→] que exigeix que sigui el client (i no pas el sistema) qui tingui la responsabilitat d'assegurar les PRE.

1.2.5.Desenvolupament sota contracte, pàgina 14

4 Contracte robust

La identificació robusta coma a cas particular. El principi de la *identificació robusta* assegura que el client té els mecanismes necessaris per a comprovar

¹³La creació només té sentit en identificació externa.

¹⁴Útils en el sentit que juntament amb el *Desenvolupament sota contracte*[→] permeten construir sistemes robusts.

1.2.5.Desenvolupament sota contracte, pàgina 14

les PRE on es fa referència a un identificador passat com a argument de l'ES. Aquest però és un cas particular del cas general: el client ha de tenir els mecanismes per assegurar o comprovar les PRE.

Contracte robust. El principi del *Contracte robust*, que anunciem tot seguit, és aquesta generalització. És un principi molt genèric, que caldrà anar perfilant. De moment, però, en fem la presentació.

Principi



Contracte robust. El sistema ha d'oferir (en la seva interfície) tots els mecanismes necessaris per tal d'assegurar que el client pot comprovar totes les PRE.

Principis i definicions del capítol

Principis

Contracte robust, 330

Identificació robusta, 329

Definicions

Identificació externa, 326

Identificació interna, 326

Identificació mixta, 326

Mecanisme de comprovació
d'identificador, 327

Recuperació exhaustiva, 323

Recuperació referencial, 324

Capítol 17

Flux comunicatiu

1	Arquitectura de tres capes	337
1.1	Disseny de tres capes	337
1.2	Especificació de tres capes	339
2	Flux comunicatiu extern	340
2.1	Principi de la <i>Comunicació amb identificadors</i>	340
2.2	Comunicació en una arquitectura per capes	341
2.3	Principi de les <i>Connexions</i>	345
2.4	Dos principis; dues comunicacions	346
3	El flux comunicatiu intern	348
3.1	El problema de la cistella	348
3.2	Comunicació de bocins	349
3.3	Comunicació de carmanyoles	350
3.4	Comunicació de targetes	350
3.5	Principi de la <i>Cistella</i>	352
4	Encaixada	352
4.1	El problema de l'encaixada	352
4.2	Resoldre l'encaixada	353
4.3	El paper del compromís en el disseny	354
4.4	L'encaixada i els repositoris centralitzats	355
	Principis	356
	Definicions	356

Contingut detallat del capítol 17

1	Arquitectura de tres capes	337
1.1	Disseny de tres capes	337
	Acoblament i cohesió dels subsistemes	337
	Responsabilitats dels subsistemes	337
	Cohesió dels subsistemes	337
	Acoblament dels subsistemes	337
	Desenvolupament per capes	337
	Una mirada endavant	337
	Profunditat il·limitada	338
	Profunditat restringida al màxim	338
	Sistema de tres capes	338
1.2	Especificació de tres capes	339
1.2.1	Del disseny a l'especificació	339
	Divideix i venç	339
	Del disseny a l'especificació	339
	Divisió del problema	339
	Redefinició del desenvolupament per capes	339
2	Flux comunicatiu extern	340
2.1	Principi de la <i>Comunicació amb identificadors</i>	340
	Necessitats de la comunicació	340
	El paper dels identificadors	340
	Un principi que ho resumeix tot	341
	Comunicació amb identificadors	341
	Un principi que no ho és	341
2.2	Comunicació en una arquitectura per capes	341
2.2.1	Identificadors i <i>Presentació</i>	341
	Què veu l'usuari	341
	Contextualització de la <i>Comunicació amb identificadors</i>	341
	Identificadors i <i>Presentació</i>	341
	Un principi que no ho és	341
2.2.2	Els ES i el desenvolupament per capes	342
	Definició d'ES	342
	Doble lectura dels ES	342
	Les capes de l'especificació	342
	La capa de <i>Presentació</i> en l'especificació	342
	La capa de <i>Domini</i> en l'especificació	342
	Comunicació entre les capes	342
2.2.3	Els paper dels ES	342
	Els ES en l'especificació	342
	El paper dels ES	342
	Llibertat d'acció	343
	Principi de les connexions	343
	Del disseny a l'especificació	343

	Abstracció de la presentació	343
	Flexibilitat de la comunicació	344
	Perquè la flexibilitat és possible	344
	Excursió. (La <i>Presentació</i> i el seu "exterior")	344
	El perquè de la notació	344
2.3	Principi de les <i>Connexions</i>	345
	Les connexions	345
	Connexions	345
	La comunicació és possible	345
	Identificadors en els ES	345
	Objectes en els ES	345
2.4	Dos principis; dues comunicacions	346
	Una comunicació desglossada	346
	Conclusions en termes planers	347
3	El flux comunicatiu intern	348
3.1	El problema de la cistella	348
	Flux comunicatiu intern	348
	Comunicació d'identificadors	348
	Comunicació en general	348
	El context de la comunicació	348
	Excursió. (Atribut)	348
	La cistella	349
	Bocins, carmanyoles i targetes	349
	Problema de la cistella	349
3.2	Comunicació de bocins	349
	Una clara justificació	349
	Violació del principi d' <i>Ocultació total</i>	349
	Acoblament de dissenys	349
3.3	Comunicació de carmanyoles	350
	Anàlisi de les carmanyoles	350
	Desacoblements del disseny intern de C	350
	Delimitació de l'acoblament de dissenys	350
	Delegació en el paràmetre	350
3.4	Comunicació de targetes	350
	Anàlisi de les targetes	350
	Cadena de delegacions	350
	Carmanyoles i targetes	351
	Necessitat de visibilitat sobre el repositori	351
	Visibilitat local.	351
	Visibilitat de paràmetre.	351
	Visibilitat d'atribut.	351
	Cadenes col·laboratives	351
	Acoblament múltiple	351
	Targetes i eficiència	352
3.5	Principi de la <i>Cistella</i>	352
	Cistella	352

El principi de la <i>Cistella</i> en termes metafòrics	352
4 Encaixada	352
4.1 El problema de l'encaixada	352
Dos principis en joc	352
Resoldre la discrepància	352
La recuperació és possible	352
Encaixada	352
4.2 Resoldre l'encaixada	353
Resolució d l'encaixada	353
Assegurem l'encaixada	353
Identificació referencial	353
<i>Identificació referencial en termes planers</i>	353
4.2.1 El controlador encaixador	353
4.2.2 Encaixador per delegació del controlador	353
Delegació de l'encaix	353
Mecanisme de la delegació de l'encaix	353
Encaixador executor	354
4.2.3 Encaixador tardà	354
Raonament fins el moment	354
Extensió del raonament	354
4.3 El paper del compromís en el disseny	354
Prioritats i compromís	354
Solidesa argumentada	355
Un principi que és més que un principi	355
Solidesa en el compromís	355
Solidesa de la teranyina	355
4.4 L'encaixada i els repositoris centralitzats	355
Simplicitat	355
Empitjorament de l'acoblament i la cohesió	355
Principis	356
Definicions	356

1 Arquitectura de tres capes

1.1 Disseny de tres capes

Acoblament i cohesió dels subsistemes. Els principis de *Baix Acoblament* i *Alta Cohesió* que estem aplicant als components també són aplicables a qual-sevol àmbit, no només als components. I en concret, també són aplicables als sistemes i subsistemes.

Responsabilitats dels subsistemes. Les responsabilitats assignades a un sistema o a un subsistema són responsabilitats de gra gruixut, que dins del subsistema en qüestió es descomposen en responsabilitats més fines.

Exemple 80 (Subsistema de bases de dades) *El subsistema Bases-DeDades té la responsabilitat de gestionar tota la informació que tenim emmagatzemada en una base de dades.*

Internament aquest subsistema consta de diversos components que tenen responsabilitats com: gestionar la connexió a la base de dades; creació de la base de dades; població inicial de les diferents taules; realitzar les diferents consultes; visualització dels resultats de les consultes;

Cohesió dels subsistemes. Per tal d'obtenir una bona *cohesió* dels subsistemes es demana que cadascun d'ells tingui una responsabilitat delimitada i ben definida.

Acoblament dels subsistemes. Per tal de disminuir l'*acoblament* entre els subsistemes es posen restriccions entre les comunicacions permeses entre ells.

Desenvolupament per capes. El desenvolupament basat en *capes* és una solució simple i elegant que ofereix una bona cohesió i un baix acoblament entre els subsistemes.

Definició. Desenvolupament per capes (primera aproximació)

El desenvolupament per capes¹ consisteix en oferir una solució software dividida en subsistemes tals que:

- **Capes apilades.** Cada subsistema constitueix un element d'una pila
- **Comunicació unidireccional.** Cada subsistema es pot comunicar amb els subsistemes que té per sota d'ell en la pila de subsistemes, però no amb els que té per sobre
- **Profunditat delimitada.** Es poden posar límits a la quantitat de capes que un subsistema pot aprofundir per poder-s'hi comunicar

Una mirada endavant. Les capes, tal i com les hem presentat, corresponen a la solució software. Per tant, tant tenen sentit en el disseny com en la implementació. Més endavant[→] veurem com la definició de capes es pot estendre a l'especificació, i d'aquesta manera abastar tot el desenvolupament.

Profunditat il·limitada. Si no posem límits a la *profunditat* tenim el que sovint s'anomena *Sistema de capes transparents*. En aquest cas, si numerem les capes des de 1 (fons de la pila) a n (cim de la pila) cada capa i es pot comunicar amb totes les capes $j \leq i$ amb $1 \leq i, j \leq n$.

Profunditat restringida al màxim. Si donada una capa i només deixem que es comuniqui amb la capa $i-1$, per a $1 \leq i \leq n$ (és a dir, cada capa només es pot comunicar en la capa de sota seu en la pila de subsistemes), llavors diem que tenim un *Sistema de capes opaques* o *Sistema de capes sòlides*.

Sistema de tres capes. Una manera molt habitual de treballar és el sistema de tres capes. Generalment es parteix d'aquesta arquitectura i, si cal, es subdivideix alguna capa o se n'introdueixen de noves. La definició que segueix és una de les múltiples possibles.

Definició. Sistema de tres capes

Sistema basat en capes opaques, que consta només de les següents capes, descrites des del cim de la pila:

- **Presentació.** Responsable de la interacció bidireccional amb l'usuari extern. Entre d'altres:
 - Interacció d'entrada. Captura de les peticions realitzades pels usuaris del sistema.
 - Interacció de sortida. Mostra resultats de les operacions; informa de qualsevol circumstància, com els errors.
- **Domini.** Responsable de les col·laboracions exigides per l'especificació
És on pròpiament es resol el problema
- **Persistència.** Responsable de la persistència de les dades

*Manntiment de les dades en bases de dades, fitxers, etc. Man-
teniment de l'estat de les sessions de treball.*

! (44)

Nosaltres treballarem amb *sistemes de tres capes*. I en concret ens focalitzem en la capa del *Domini*. Per tant els dissenys que desenvolupem són principalment dissenys de la capa del *domini* del sistema que presentem com a solució del problema.

1.2 Especificació de tres capes

1.2.1 Del disseny a l'especificació

Divideix i venç. La partició del problema de partida en problemes més simples facilita tant la descripció del problema com la cerca d'una solució.

Del disseny a l'especificació. El disseny proposa una descomposició de la solució en capes. Però cada capa és un subsistema que resol una problemàtica determinada del problema. Per tant, cada capa dels disseny resol un subproblema del problema inicial.

Divisió del problema. Dividir el problema inicial en problemes més simples, no només no és perjudicial, sinó que pot ser força avantatjós. I el disseny proposa una divisió de la solució que indueix una divisió del problema. Així, de la necessitat del disseny en fem una virtut de l'especificació: el problema l'especifiquem també per capes.

Definició. Especificació per capes

- **Presentació.** *Ha de definir els mecanismes d'interacció, els formats de presentació de dades, el sistema de notificacions, etc.*

Cal expressar, per exemple, si una galeria de fotos es presenta en forma de graella o de llista; però no cal indicar la mida de cada casella de la graella, la quantitat de caselles en una fila, o l'espaiat entre les diferents caselles.²

- **Domini.** *Informació que manipula el problema, i la seva manipulació principal.*

L'especificació del domini és el que fins ara anomenàvem especificació

- **Persistència.** *És una responsabilitat pròpia de la solució, i del tot aliena al problema*

Redefinició del desenvolupament per capes. El *desenvolupament per capes* → és aplicable al domini del problema. Això permet redefinir el desenvolupament per capes per tal que englobi qualsevol etapa del desenvolupament.

Definició pàgina 337

²La definició exacta de què conté l'especificació de la capa de *Presentació*, i quines decisions són més pròpies del disseny i de la implementació queden fora de l'abast d'aquest treball. Aquí n'hi ha prou amb recordar la diferència bàsica entre especificació i disseny: en l'especificació de la *capa de presentació* cal indicar *quines* interaccions i visualitzacions són necessàries; en el disseny de la *capa de presentació* cal decidir *com* es realitzaran aquestes interaccions i visualitzacions.

Definició. Desenvolupament per capes (proposta definitiva)

El desenvolupament per capes consisteix en dividir tant el problema com la solució en subsistemes tals que:

- *Capes apilades.* Cada subsistema constitueix un element d'una pila
- *Comunicació unidireccional.* Cada subsistema es pot comunicar amb els subsistemes que té per sota d'ell en la pila de subsistemes, però no amb els que té per sobre.
- *Profunditat delimitada.* Es poden posar límits a la quantitat de capes que un subsistema pot aprofundir per poder-s'hi comunicar

2 Flux comunicatiu extern

2.1 Principi de la Comunicació amb identificadors

Necessitats de la comunicació. A 14.Encontres en la tercera fase, pàgina 253,[→] hem analitzat la problemàtica d'establir comunicació entre dos mons diferents:

- **Problema comunicatiu.** Els dos mons que volem comunicar són el problema i el sistema; la realitat i el model; l'especificació i el disseny. I no es tracta pas de tres problemes comunicatius diferents, ans d'un de sol plantejat des de perspectives diferents.[→]
- **Llenguatge de comunicació.** La comunicació només és possible si es disposa d'un llenguatge vàlid en ambdós mons, i que compleixi unes determinades propietats.[→]
- **Llenguatge de noms.** El conjunt de noms del model construït segons la *Modelització contínua* és un llenguatge que compleix amb les propietats exigides per a la comunicació.[→]

El paper dels identificadors. A 15.Identificadors, pàgina 277,[→] hem vist com les *propietats d'identificació* són un mecanisme que proporcionen noms a les realitzacions dels components. Aquests noms són el que anomenem *identificadors*:

- **Mons a dos nivells.** La presència d'abstraccions (conceptes i components, entre d'altres) i de les seves realitzacions (entitats i objectes, respectivament) estructuren el món en dos nivells diferents. El llenguatge de comunicació, per tant, també és a dos nivells.[→]
- **Identificació.** Les *propietats d'identificació* són un mecanisme que proporciona nom a les realitzacions dels conceptes i components.[→] Si els components tenen unes determinades propietats,[→] llavors els seus *identificadors*, juntament amb els noms dels components, constitueixen un llenguatge de noms amb les propietats necessàries per a la comunicació.

- **Delimitació de la identificació.** N'hi ha prou en proporcionar propietats d'identificació pels components per assegurar que tota realització (del model o de la realitat) és identificable. →

Un principi que ho resumeix tot. El principi de la *Comunicació amb identificadors* resumeix totes les conclusions anteriors:

Comunicació amb identificadors. L'única comunicació vàlida entre l'especificació i el disseny, entre realitat i model, són els valors d'identificació dels components ben definits

Principi



Un principi que no ho és. El principi de la *Comunicació amb identificadors* pròpiament és una necessitat, i no pas una proposta. Per tant s'ha de considerar com una **regla** més que com a **principi**.³

2.2 Comunicació en una arquitectura per capes

2.2.1 Identificadors i Presentació

Què veu l'usuari. L'usuari extern al sistema només té coneixement de la capa de *Presentació* del sistema. Per tant la comunicació externa, la comunicació entre els dos móns irreconciliables, és la comunicació entre l'exterior del sistema i la seva capa de *Presentació*.

Contextualització de la *Comunicació amb identificadors*. Per tant, en una arquitectura per capes, es pot reformular el *Comunicació amb identificadors* en els següents termes:

Identificadors i Presentació.

- La comunicació entre el sistema i el seu exterior es redueix a la comunicació entre la *Presentació* i l'exterior del sistema
 - Aquesta comunicació només és possible mitjançant els identificadors
-

Principi



Un principi que no ho és. La decisió de desenvolupar amb capes o no és justament una decisió, recomanable però no exigible. Però un cop presa la decisió la comunicació queda restringida a la capa de *Presentació*; i aquesta comunicació només pot ser a través dels *identificadors*.

³La presentació en forma de principi és per facilitar-ne el seu accés: apareixerà en els índexs de principis. A tots els efectes, però, és una *regla* que no admet violacions. A 1.3.Regles i principis, pàgina 16 → es presenta la diferència entre *regla* i *principi*.

2.2.2 Els ES i el desenvolupament per capes

2.1.1. Què ofereix l'especificació, pàgina 22

Definició d'ES. A 2.1.1. Què ofereix l'especificació, pàgina 22, → hem definit el ES com les peticions externes *per sota* de la capa de *Presentació*. Per tant en aquesta definició ja hi havia implícita la idea del desenvolupament a tres capes:

(69)

Els ES comuniquen la *Presentació* amb el *Domini*

2.1.2. Lectura dels contractes dels ES, pàgina 22

Doble lectura dels ES. A 2.1.2. Lectura dels contractes dels ES, pàgina 22, → hem analitzat com els ES es poden llegir tant des del problema com des de la solució.

17.1.2. Especificació de tres capes, pàgina 339

Les capes de l'especificació. L'*especificació per capes* l'hem introduïda per subdividir la complexitat del problema, i alhora fer-ho de manera simètrica a la divisió introduïda pel *disseny per capes*. →

La capa de *Presentació* en l'especificació. La capa de *Presentació* de l'especificació explica, en termes del problema, *com* l'usuari extern interactua amb el sistema; *com* li demana les coses i *com* n'obté el resultat.

Aquest *com* és un *com* força abstracte. Per exemple: necessitem una pantalla amb tal informació, però no baixem al detall del tipus de lletra, mida o color. Aquesta abstracció és la que ens permet afirmar que estem especificant (definint el *què*)⁴ i no pas dissenyant o implementant (definint el *com*)

La capa de *Domini* en l'especificació. La capa de *Domini* de l'especificació defineix les funcionalitats del sistema i les interrelacions entre la informació del problema.

Comunicació entre les capes. Resumim el que hem vist fins ara:

- L'usuari extern es comunica només amb la *Presentació*
- Les peticions sobre el *Domini* les fa la *Presentació*

2.2.3 Els paper dels ES

Els ES en l'especificació. Des de l'especificació els ES es poden veure com un mecanisme que ens permet abstraure'ns de la capa de presentació concreta. Ens és igual si ens donen la informació fent seleccions en un desplegable, si ens envien un mail, o escanegem un codi de barres. En qualsevol d'aquestes circumstàncies la informació necessària és la mateixa.

⁴Ep!!!! En que quedem, definim el *què* o el *com*? La resposta és subtil: definim el *què* del *com*. La subtilesa és similar a la de la modelització a múltiples nivells: el que en un nivell és un model d'una realitat; en el nivell superior és la realitat a modelitzar. Aquí passa el mateix: dins l'especificació per capes la presentació és el *com* de les peticions; quan considerem aïlladament la presentació aquesta explica un *què* que cal materialitzar amb un *com*: per això diem que és el *què* (està en termes del problema) del *com* (descriu, en el problema, els mecanismes d'interacció).

El paper dels ES. Un cop construïda una solució per a un problema (és a dir, un cop construït el sistema) la comunicació amb l'exterior serà responsabilitat exclusiva de la *presentació*. Per tant els ES no formen part d'aquesta comunicació entre dos mons.

El paper dels ES

- Els ES permeten *focalitzar* en les necessitats del problema sense tenir en compte les mecanismes concrets d'interacció amb l'usuari; són un mecanisme d'*abstracció*
- Amb la solució al problema construïda, la *comunicació* entre sistema i usuari és responsabilitat exclusiva de la *Presentació*

(70)

Llibertat d'acció. De la conclusió anterior s'extreu que *comunicació* entre *Presentació* i *Domini* és una comunicació dins d'un mateix món, i per tant no ha d'estar restringida per les condicions del principi de la *Comunicació amb identificadors*. →

Principi, pàgina 341

Principi de les connexions. A 17.2.3.Principi de les *Connexions*, pàgina 345, → es presenta el principi de les *Connexions*, que diu quins poden ser els arguments de l'ES, des de la perspectiva del disseny.

17.2.3.Principi de les *Connexions*, pàgina 345

Del disseny a l'especificació. Si partim d'un model correcte⁵ tot objecte del disseny és el model d'una entitat de la realitat; i tot identificador interpretable en el disseny també és interpretable en la realitat.⁶ Com a conseqüència el principi de les *Connexions* també explica quins són els arguments vàlids pels ES de l'especificació.

Avantatges de desenvolupar amb ES.

Els ES permeten:

- **Abstracció de la Presentació.** Especificar i dissenyar sense tenir en compte la interacció concreta de l'usuari
- **Flexibilitat en la comunicació.** Els arguments dels ES no necessàriament han de ser *identificadors* ja que els ES no es corresponen a una comunicació entre dos mons

(71)

⁵Les condicions exactes de què significa exactament un model *correcte* per a la comunicació les hem analitzades a 14.3.Mons comunicables, pàgina 259. →

14.3.Mons comunicables, pàgina 259

⁶Això és conseqüència d'haver construït el model a partir de components correctes". La definició exacta de què és un component correcte per a la comunicació la tenim a 15.4.Component ben definit, pàgina 296. →

15.4.Component ben definit, pàgina 296

Abstracció de la presentació. L'*abstracció de la presentació* és coherent amb la idea de modelització: focalitzem en els aspectes que ens interessin de la realitat. Els ES permeten descriure un *model funcional* del problema. (La presentació correspondria al *model d'interacció*).

Flexibilitat de la comunicació. En especificar podem donar entitats com a arguments dels ES; aquestes entitats caldrà que es corresponguin a un dels objectes del nostre model. I a la inversa, en dissenyar podem considerar objectes com a arguments dels ES; i caldrà que aquests objectes siguin el model d'alguna entitat de la realitat. En conclusió, no estem encotillats en els *identificadors*.

Perquè la flexibilitat és possible. Els ES es descriuen només en un dels mons, tot i que són interpretables en els dos mons. Per això tant li fa si usem objectes o entitats. Un cas a part són les peticions que des de l'exterior es fan a la *Presentació*: aquí hi ha comunicació entre dos mons, i per tant cal usar els *identificadors*.

Excursió. (La *Presentació* i el seu "exterior") En *especificació* la *Presentació* comunica el problema amb el context d'aquest problema. Potser tant el problema com el seu context pertanyen a una mateixa realitat, i per tant la comunicació entre ells és possible sense la necessitat del context. Malgrat això no en podem tenir la certesa absoluta, i per tant és preferible, per simplicitat i per simetria amb el disseny, considerar que el problema i el seu context són realitats distintes que es poden comunicar mitjançant identificadors.

En el *disseny* la *Presentació* comunica el problema amb la seva solució; l'exterior del sistema amb el propi sistema. Per tant es tracta de dos mons totalment diferents, i necessàriament caldrà emprar els identificadors per a la comunicació.



Notació. A partir d'ara, quan parlem de *sistema* ens referim al *Domini* del sistema

El perquè de la notació. A partir d'ara parlarem de *sistema* i els seu *exterior* quan pròpiament hauríem de parlar de la *Presentació* i el *Domini*. Aquesta simplificació es justifica a partir de les següents consideracions:

- Els ES permeten abstraure'ns de la presentació. El resultat és una simplificació en tot el desenvolupament
- La nostra tasca està focalitzada en el disseny. Si ens abstraiem de la presentació, el que queda és el *Domini* com a responsable de totes les funcionalitats del disseny
- Resultat de tot plegat és que el sistema que estem desenvolupant (o el problema que estem resolent) és el *Domini*; i el seu exterior és la *Presentació*

2.3 Principi de les Connexions

Les connexions. Els ES són el mecanisme de comunicació entre el sistema i l'exterior;⁷ constitueixen el conjunt de *connexions* del sistema amb l'exterior.⁸

Connexions. Els arguments dels ES han de ser:

- Identificadors especulars⁹ amb recuperació referencial \rightarrow , \rightarrow dins el sistema
- Objectes amb recuperació exhaustiva \rightarrow visible en la interfície del sistema

Principi



Definició pàgina 294

pàgina 324

Definició pàgina 323

La comunicació és possible. El principi de les *Connexions* diu com s'han de *connectar* el món exterior i el sistema: les connexions només són possibles amb identificadors tals que els seus valors són comprensibles en ambdós mons; o amb objectes que prèviament hem obtingut del sistema, segurament a partir dels seus identificadors.¹⁰

Identificadors en els ES. En el cas dels identificadors, l'exigència que es tracti d'*identificadors especulars* \rightarrow assegura que les peticions exteriors seran convenientment enteses pel sistema; l'exigència de la *recuperació referencial* \rightarrow assegura que el sistema té els mecanismes necessaris per recuperar l'objecte identificat per l'argument de l'ES.

Definició pàgina 294

Definició pàgina 324

Objectes en els ES. En cas de no usar identificadors, el principi de les *Connexions* diu que només es poden usar realitzacions de components com a arguments dels ES en el cas que el sistema ofereixi els mecanismes per tal que els usuaris del sistema puguin obtenir aquestes realitzacions. Però com que les realitzacions de components pertanyen al món de la solució software, els agents externs del sistema (els actors) no els poden crear (ni tant sols usar: per a ells no tenen cap mena de sentit). Si de cas els actors poden demanar al sistema que els doni una realització, amb l'única finalitat d'usar-la com a mecanisme comunicatiu amb el sistema.

Exemple 81 *Sigui l'ES novaInscripcio(caminada). Què és caminada?*

Pot ser un identificador de Caminada:

caminada : $Id_{Caminada}$ ¹¹

⁷Més concretament, entre la *Presentació* i el *Domini*.

⁸Més concretament, entre la *Presentació* i el *Domini*.

⁹La definició precisa d'*identificador especular* la trobem a 3.5., pàgina 294. \rightarrow Si no volem entrar en els detalls formals n'hi ha prou en dir que tenim un identificador que tant ens serveix per a identificar en el model com en la realitat.

Definició pàgina 294

¹⁰La recuperació a través de l'identificador no és l'únic mecanisme. Per exemple, podem tenir un ES que retorni la *i:Inscripció activa*.

O pot ser una realització de Caminada.

caminada : Caminada

*En aquest darrer cas el principi de les Connexions exigeix que el sistema ha d'oferir en la seva interfície un mecanisme per la recuperació exhaustiva de les caminades. Per exemple, podem dotar al sistema d'un nou ES que donat un identificador de Caminada retorni la realització corresponent de Caminada:*¹²

obtenirCaminada (id_{Caminada}) : Caminada

I si caminada és un identificador? Llavors el principi de la Identificació robusta[→] exigeix que hi hagi un mecanisme de comprovació d'identificador per a la Caminada:

existeixCaminada (id_{Caminada}) : Bool

Cal observar també que si usem un mecanisme de recuperació referencial també es compleixen les condicions que activen el principi de la Identificació robusta.

Tot plegat ho resumim en la següent taula, $A \rightarrow B$ indica que A exigeix B; i on $A \dashrightarrow B$ indica que un cas particular del que exigeix A és B:

<i>caminada identificador</i>	\rightarrow	<i>existeixCaminada (id_{Caminada}) : Bool</i>
<i>caminada objecte</i>	\dashrightarrow	<i>obtenirCaminada (nom) : <c:Caminada></i>
<i>obtenirCaminada (...)</i>	\rightarrow	<i>existeixCaminada (id_{Caminada}) : Bool</i>

16.3.3. Identificació robusta, pàgina 329

14.2. Un problema comunicatiu, pàgina 257

(72)

Problema comunicatiu resolt

El principi de les Connexions estableix el mecanisme necessari per resoldre els problemes comunicatius expressats en la pregunta 34, pàgina 258 (apartat 14.2. Un problema comunicatiu, pàgina 257).[→]

2.4 Dos principis; dues comunicacions

Una comunicació desglossada. El problema de la comunicació entre l'exterior i el sistema, com que estem desenvolupant per capes, el podem desglossar en dues comunicacions:

Exterior	\longleftrightarrow	Presentació
Presentació	\longleftrightarrow	Domini

¹¹La notació *caminada*: *Id_{Caminada}* pot fer pensar que *Id_{Caminada}* és un component, però res més lluny de la realitat: *Id_{caminada:Caminada}* s'ha d'entendre com una notació per expressar un identificador de Caminada.

¹²El principi del es Connexions exigeix una recuperació exhaustiva. Aquí proposem un mecanisme de recuperació referencial, que és un cas particular de la recuperació exhaustiva.

Connexions i capes

(73)

- El principi de la *Connexions* diu com són els arguments dels ES
Explica com és la comunicació entre la capa de *Presentació* i la capa de *Domini*
- El principi de la *Comunicació amb identificadors* diu com és la informació comunicada des de l'exterior del sistema
Diu com és la comunicació entre l'usuari extern i la capa de *Presentació*

Conclusions en termes planers. A efectes pràctics, el que diuen els principis de la *Comunicació amb identificadors* i de les *Connexions* queda sumariat en la següent conclusió:

Aspectes de comunicació en el desenvolupament.

(74)

- En especificar l'aspecte dinàmic del problema (el model dels casos d'ús) ens hem d'oblidar de **com** l'usuari proporciona o obté la informació necessària
- El que ens preocupa és **quina** és aquesta informació
- Podem passar **identificadors**. Serà responsabilitat del disseny assegurar que disposem de:
 - Mecanismes **interns (o no)** per obtenir l'objecte corresponent. (Recuperació referencial exigida pel principi de les *Connexions*)
 - Mecanismes **externs** per comprovar si en el sistema existeix un objecte amb aquest identificador. (Comprovació d'identificador exigida pel principi de la *Identificació robusta*)
- Podem passar **entitats**. Serà responsabilitat del disseny assegurar que disposem de **mecanismes externs** per obtenir l'objecte que modelitza aquesta entitat. (Recuperació exhaustiva exigida pel principi de les *Connexions*)

L'objecte obtingut pròpiament no es retorna a la realitat, ans queda ancorat en els llimbs de la *Presentació*. Però aquesta també tindrà el seu correlat en termes del model. En d'altres paraules, l'objecte software passa d'una capa de software (el *Domini* a una altra la *Presentació*)

3 El flux comunicatiu intern

3.1 El problema de la cistella

Flux comunicatiu intern. Els arguments dels ES són la informació que el sistema¹³ necessita per resoldre la responsabilitat que li demana l'ES. Per resoldre aquesta responsabilitat necessita que diferents objectes col·laborin entre sí, i per tant la informació rebuda en els arguments de l'ES ha de comunicar-se a través d'aquests objectes.

Comunicació d'identificadors. Què passa quan l'argument de l'ES és un identificador? Quina informació es transmet internament, el propi identificador, o l'objecte recuperat a través d'aquest?

Comunicació en general. Aquesta qüestió és un cas particular d'una qüestió més general:

? (45)

Internament, el sistema quin tipus d'informació manipula per comunicar els diferents objectes? Quin tipus d'informació "viaja" entre els diferents objectes?

El context de la comunicació. Suposem que un determinat component R rep una determinada responsabilitat expressada pel missatge $m()$. Suposem que per exercir aquesta responsabilitat li cal accedir a la informació continguda en els atributs x i y d'una realització de $c:C$. Suposem que les realitzacions de $c:C$ s'identifiquen amb l'atribut id . Quins són els arguments del missatge $m()$: l'identificador de la realització de $c:C$, el propi $c:C$, o la informació mínima imprescindible expressada pels atributs x i y de C ?

Excursió. (Atribut) Entenem per *atribut* d'un element una propietat de l'element que es pot conèixer observant el propi element.¹⁴ En el cas dels objectes, és una propietat de l'objecte el valor de la qual forma part del propi objecte.

La *visibilitat d'atribut* és un *atribut*; però no tot atribut és una visibilitat d'atribut. El valor de la visibilitat (la destinació de l'enllaç dirigit) no forma part de l'objecte, sinó d'un enllaç dirigit del qual l'objecte n'és l'origen.

En els apartats 10.11.2.1.La informació en el MC i 10.11.2.2.La informació a MComp (pàgina 192 i següents)^{→,→} s'ha analitzat detingudament el concepte d'*atribut*.

A 10.11.2.2.La informació a MComp, pàgina 194,[→] s'ha introduït el terme *propietat atributiva* justament per evitar confusions quan parlem d'*atribut* com a propietat, i com a abstracció dels valors interns d'un objecte.

¹³Pròpiament, la capa de *Domini* del sistema

¹⁴*Diccionari General de la Llengua Catalana*: Atribut. Qualitat o manera d'ésser considerada com a pròpia d'algú o d'alguna cosa

Davant la necessitat de comunicar determinada informació a una realització, quina és la informació que realment s'ha de passar:

(46) ?

- *La informació mínima imprescindible.* En l'exemple, $m(x, y)$.
- *L'objecte que conté aquesta informació.* En l'exemple, $m(c:C)$.
- *L'identificador d'aquest objecte.* En l'exemple, $m(id:Id_C)$.

La cistella. La pregunta 46 es pot expressar en termes de la metàfora de la *cistella*. Entenem per *cistella* la informació que anem fent circular pel sistema. La cistella pot portar *bocins*, *carmanyoles*, i *targetes*.

Bocins, carmanyoles i targetes. Quan tenim gana (necessitem informació) podem picotejar els bocins que portem. O podem obrir les carmanyoles per picar dels bocins que contenen. O podem mirar l'adreça de les targetes que ens indiquen les botigues on podem comprar carmanyoles plenes de menjar.

Problema de la cistella. El *problema de la cistella* (equivalent a la pregunta 46) llavors es pot expressar com:

Què ens convé més portar a la cistella:

(47) ?

- *Bocins.* En l'exemple, $m(x, y)$.
- *Carmanyoles.* En l'exemple, $m(c:C)$.
- *Targetes.* En l'exemple, $m(id:Id_C)$.

3.2 Comunicació de bocins

Una clara justificació. Estem en el cas en què la *cistella* porta la informació mínima indispensable per exercir la responsabilitat demanada. Per tant sembla que és l'opció correcta. (Quines ganes d'obrir carmanyoles; agafem el bocí de fuet i queixalada directa).

Violació del principi d'Ocultació total. Cal observar però, que per aconseguir els bocins cal haver demanat a $c:C$ que ens els proporcioni. Això significa que cal dotar la interfície del component C dels mecanismes per comunicar tota aquella informació que pugui ser útil en qualsevol necessitat comunicativa. Pel missatge $m()$ necessitem x i y ; però potser per un missatge $m_2()$ necessitem z . En d'altres paraules, C exposa, a través de la seva interfície, tota (o gran part) de la seva informació interna. L'*Ocultació total* \rightarrow ha desaparegut.

Acoblament de dissenys. El missatge $m()$ necessita la informació continguda a x i y , però li és indiferent si aquesta informació està a x i y o en una altra banda. Si en un futur el disseny de C canvia, i com a resultat distribuïm la informació que $m()$ necessita en tres atributs (p , q i r) enlloc de dos, caldrà fer molts canvis. Caldrà canviar l'emissor del missatge $m()$ per tal que demani a $c:C$ els nous atributs; i caldrà canviar R per tal que realitzi la responsabilitat de $m()$ usant p , q i r enlloc de x i y . El problema és que el disseny de R i la seva interfície estan *acoblats* al disseny intern de C (com a conseqüència directa de la seva no *Ocultació total*). I aquest acoblament dificulta els canvis (per manteniment, millores o extensions) de C .

3.3 Comunicació de carmanyoles

Anàlisi de les carmanyoles. Estem en el cas en què la *cistella* porta l'objecte que conté la informació mínima indispensable per exercir la responsabilitat demanada.

Desacoblements del disseny intern de C . En aquest cas el principi d'*Ocultació total* \rightarrow sobre C es manté. L'emissor del missatge $m()$ no s'ha de preocupar dels possibles canvis en la interfície de C , ja que sempre farà la mateixa petició: $m(c)$. Per tant la interfície de R també es manté inalterable davant dels canvis de la interfície de C . Com a conclusió podem dir que la interfície de R i el disseny de l'emissor de $m()$ s'han desacoblat del disseny intern de C .

Delimitació de l'acoblament de dissenys. La petició que R ha de fer a C per obtenir-ne els *bocins* necessaris és una esclatxa a l'*Ocultació total*; i *acobla* R al disseny intern de C . A diferència del cas de *comunicació de bocins* ara l'acoblament afecta internament a R , però no la seva interfície; per tant l'acoblament de dissenys no afecta l'exterior de R . Conclusió: amb la *comunicació de carmanyoles* el disseny obtingut és més estable, modificable i mantenible.

Delegació en el paràmetre. Si malgrat tot l'acoblament que R té sobre C no ens convenç, sempre podem *delegar en el paràmetre*. La idea és que enlloc d'extreure la informació necessària de la carmanyola, li demanem a ella que faci la tasca.¹⁵

En el cas de l'exemple, la delegació en el paràmetre es pot descriure de la següent manera: una realització de R rep el missatge $m()$ amb l'argument $c:C$; en el disseny de R , i per tal d'evitar publicar parcialment C , deleguem a $c:C$ la part de la responsabilitat de R que depèn de la informació de $c:C$.

3.4 Comunicació de targetes

Anàlisi de les targetes. Estem en el cas en què la *cistella* porta l'identificador de l'objecte que conté la informació mínima indispensable per exercir la responsabilitat demanada.

¹⁵ Aquí la metàfora grinyola. És com si li demanéssim a la carmanyola que mengi per nosaltres!

Cadena de delegacions. Suposem una situació en la que la responsabilitat de $m()$ es va delegant al llarg de tota una cadena de delegacions. Cada objecte de la cadena, a excepció del primer i el darrer, es limiten a delegar la responsabilitat i a comunicar la informació rebuda; són simples ponts. El primer objecte de la cadena (l'*iniciador*) és qui omple la cistella (en aquest cas, obté l'identificador); el darrer objecte de la cadena (l'*executor*) és qui exerceix la responsabilitat.

Carmanyoles i targetes. En una primera ullada la comunicació de targetes sembla equivalent, en termes d'ocultació i acoblament, a la comunicació de carmanyoles. La diferència és que ara l'*executor*, primer ha d'usar la targeta per obtenir la carmanyola.

Necessitat de visibilitat sobre el repositori. Però les coses no són exactament així. L'*executor* necessita conèixer els objectes de la cistella, sigui per extreure'n la informació necessària, sigui per aplicar-los-hi la delegació en el paràmetre. Per tant l'*executor* necessita recuperar els objectes que li arriben en termes d'identificadors. I per tant ha de tenir visibilitat del repositori de les realitzacions de C .¹⁶

Visibilitat local. La visibilitat local és totalment absurda per un repositori: la vida d'un repositori ha de ser superior a la de la resposta a un missatge.

Visibilitat de paràmetre. La visibilitat de paràmetre significaria que en la cadena de delegació a part d'enviar l'identificador pertinent també caldria enviar l'objecte repositori pertinent. És a dir, per evitar enviar l'objecte (la carmanyola) ens veiem obligats a enviar un multiobjecte que el conté, més l'identificador dins del multiobjecte (i ens caldrà executar el `find` dins l'*executor*)!!. Hi hem guanyat ben poc.

Visibilitat d'atribut. Només queda la visibilitat d'atribut. Per tant, l'*executor* està *acoblat*, amb visibilitat d'atribut, al repositori dels objectes que cal recuperar amb els identificadors rebuts. Com a conseqüència la *cohesió* baixa: l'*executor* a part de les responsabilitats assignades, té la responsabilitat de conèixer l'emmagatzemament (el repositori) dels objectes que necessita per treballar.

Cadenes col·laboratives. Sovint les responsabilitats demanades exigeixen llargues cadenes de missatges col·laboratius. En aquests casos una mateixa informació pot ser enviada a objectes diferents, per tal que cadascun realitzi una part de la responsabilitat demanada. El resultat és que tenim més d'un *executor*, cadascun amb una responsabilitat diferent, però que actuen sobre la mateixa informació.

¹⁶Si el repositori no és centralitzat cal, o bé tenir visibilitat de tots els fragments del repositori distribuït (i així tenim un repositori centralitzat de repositoris o fragments!), o bé alguna propietat del disseny ha d'assegurar que el repositori del qual tenim visibilitat és el que ha de contenir l'objecte a recuperar.

Acoblament múltiple. La comunicació amb targetes, en el cas de cadenes col·laboratives, significa que *cada executor* està acoblat, via visibilitat d'atribut, al repositori.

Targetes i eficiència. El problema de les cadenes col·laboratives amb targetes no és només un problema d'acoblament i de cohesió. És també un problema d'eficiència: cada *executor* ha de fer el *find* sobre el repositori. Recordem que el principi de l'*Eficiència en segon pla* → diu que no s'ha de prendre, de moment, cap decisió per motius d'eficiència; i si de cas només podem usar criteris d'eficiència per decidir entre dues alternatives igualment vàlides segons els principis de disseny. En aquest cas, però, els problemes d'eficiència reforcen la proposta negativa que sobre aquest model ens donen els principis de *Baix acoblament* i *Alta cohesió*.

Principi, pàgina 228

3.5 Principi de la Cistella

Principi



Cistella. Els arguments dels missatges seran sempre els *objectes* que continguin la informació necessària per respondre al missatge.

El principi de la Cistella en termes metafòrics. El principi de la *Cistella* diu que la cistella d'informació conté carmanyoles; res de bocins ni de targetes. Només carmanyoles.

4 Encaixada

4.1 El problema de l'encaixada

Principi, pàgina 345

Principi, pàgina 352

Dos principis en joc. El principi de les *Connexions* → diu que els ES poden contenir identificadors com a arguments; el principi de la *Cistella* → diu que internament s'han de manipular objectes, no pas identificadors.

Resoldre la discrepància. L'aplicació d'ambdós principis exigeix que un cop rebuda una informació des de l'exterior en forma d'identificador, abans de comunicar-la per dins del sistema cal recuperar l'objecte pertinent.

La recuperació és possible. Aquesta recuperació és possible justament per l'exigència en el principi de la *Cistella* que el sistema tingui mecanismes de *recuperació referencial* pels identificadors usats.

Principi



Encaixada. La primera responsabilitat que cal resoldre en el moment de rebre un ES és la de la *recuperació* dels objectes indicats mitjançant identificadors en els arguments de l'ES.

4.2 Resoldre l'encaixada

Resolució d l'encaixada. El principi de l'*Encaixada* diu que en rebre un ES el primer que cal fer és "reconvertir" tots els identificadors rebuts en els objectes que identifiquen. Aquesta reconversió és el que anomenem *resoldre l'encaixada*.

Definició. Resolució de l'encaixada

Donat un identificador com a argument d'un ES, diem que en resolem l'encaixada que n'obtenim l'objecte que referencia.

Notació. Sovint enlloc de dir que *resolem l'encaixada* d'un identificador *id*, per abreujar direm que *encaixem id*



Assegurem l'encaixada. El principi de les *Connexions* permet que els Es tinguin identificadors com a arguments; i el principi de l'*Encaixada* exigeix accedira al més aviat possible a l'objecte identificat. Per tant ens cal assegurar que la resolució de l'encaixada serà sempre possible.

Identificació referencial. El sistema disposa de mecanismes de *recuperació referencial* per tots i cadascun dels identificadors usats

Principi



Identificació referencial en termes planers. El que diu el principi del a *Identificació referencial* és que sempre que usem un identificador és perquè realment aquest referencia algun objecte. És a dir, que els identificadors usats compleixen amb el seu propòsit.

Qui és el responsable de resoldre l'encaixada?

(48) ?

4.2.1 El controlador encaixador

El responsable de capturar un ES de sistema és el seu *controlador*. Per tant sembla que aquest pot ser un bon candidat per a resoldre l'encaixada. Però llavors es viola el principi de l'*Encarrilament* \rightarrow : el controlador a part d'encarrilar resol l'encaixada.

Principi, pàgina 115

4.2.2 Encaixador per delegació del controlador

Delegació de l'encaix. L'alternativa al controlador encaixador és que el controlador delegui en algú (diguem-ne l'*encaixador*) la responsabilitat de resoldre l'encaixada.

Mecanisme de la delegació de l'encaix. Suposem una situació amb un ES amb un sol identificador com a argument, corresponent a una realització d'un component C ; i suposem que la responsabilitat demanada exigeix enviar un missatge $m(c)$ a un objecte $x:X$ (l'executor), on c és l'objecte a què fa referència l'identificador rebut. Amb aquests supòsits, el controlador delega en l'encaixador la tasca de recuperar (encaixar) $c:C$, i un cop recuperat $c:C$ el controlador envia el missatge $m(c)$ a $x:X$. El resultat és que tant el controlador, com l'encaixador, com l'executor estan acoblats a C .

Encaixador executor. La tasca de l'encaixador és similar a la d'un creador: el creador obté un nou objecte; l'encaixador obté un objecte pre-existent. Els mateixos raonament fets en parlar dels creadors[→] es pot fer amb els encaixadors. En concret, una manera de disminuir l'acoblament és fer que sigui el propi encaixador qui envii el missatge $m()$ a l'executor. El resultat és que el controlador deixa d'estar acoblat a C ; però la cohesió de l'encaixador ha baixat.

9.2.El problema de la patata calenta, pàgina 135

4.2.3 Encaixador tardà

Raonament fins el moment. La idea inicial era que fos el controlador qui resolgués l'encaixada. Degut als problemes del disseny resultant l'hem anat modificant fins arribar a un disseny on el controlador delega en l'executor tant la responsabilitat del missatge $m()$ com la recuperació dels objectes necessaris per realitzar aquesta responsabilitat.

Extensió del raonament. Si el que tenim és una cadena de delegació (o fins i tot una cadena col·laborativa) la temptació és aplicar el mateix raonament i delegar en l'executor la resolució de l'encaixada. És a dir, podem anar posposant la resolució de l'encaixada; però llavors violem el principi de la *Cistella*.[→]

Principi, pàgina 352

4.3 El paper del compromís en el disseny

Prioritats i compromís. Com veiem no hi ha una solució clara de cara a determinar el bon encaixador. Això vol dir que cal definir prioritats i cercar el compromís. En concret, en general donarem una prioritat alta al principi de la *Cistella* (simplifica el disseny i facilita les modificacions); cercarem una solució que no doni un acoblament molt fort o una cohesió molt baixa (cercarem un model equilibrat, no necessàriament òptim en aquests aspectes); i admetrem que els controladors puguin resoldre encaixades (sempre i quan la cohesió o l'acoblament no es vegin excessivament afectats).

(75)

Criteris ordenats per prendre decisions.

1. *Cistella*.[→]

Prioritat alta

2. *Alta Cohesió i Baix Acoblament*.^{→,→}
Cercarem una solució equilibrada

Principi, pàgina 352

Principi 8.2, pàgina 127

3, pàgina 128

3. Encaixades en el controlador.

Admetem com a vàlida aquesta violació de
l'Encarrilament →

Principi, pàgina 115

Solidesa argumentada. El disseny perfecte no existeix. El que hi ha són dissenys en els que es poden argumentar de manera sòlida i consistent totes i cadascuna de les decisions preses.

Principi



Un principi que és més que un principi. El principi de la *Solidesa argumentada* no és pròpiament un principi, en el sentit que no presenta una recomanació de disseny davant un context determinat. És un *metaprincipi*: expressa com s'ha de presentar el disseny: no n'hi ha prou en presentar el model resultant; cal justificar, en termes dels principis de disseny, com s'hi ha arribat.¹⁷

Solidesa en el compromís. La necessitat de prendre prioritats i cercar compromisos exigeix que quedi ben documentada quina prioritat s'ha mantingut i quines propietats intenten maximitzar o minimitzar els compromisos presos.

Solidesa de la teranyina. Aquesta necessitat d'explicitació adquireix la seva rellevància davant el principi de la *Reconsideració*: → en el moment de reconsiderar una decisió cal ser consistents amb les prioritats establertes i amb els objectius perseguits en els compromisos presos. D'aquesta manera es reforça el principi de la *Teranyina*: → el disseny és una xarxa de decisions densa i feble.

2.4.2.Reconsideració d'una assignació, pàgina 26

2.4.3.Una densa i fràgil teranyina de decisions, pàgina 27

4.4 L'encaixada i els repositoris centralitzats

Simplicitat. L'existència de repositoris centralitzats dels objectes que cal recuperar en la resolució de l'encaixada simplifica el disseny. N'hi ha prou en què l'encaixador tingui visibilitat d'atribut sobre els repositoris centralitzats corresponents.

Empitjorament de l'acoblament i la cohesió. Per contra, sota el supòsit que demanem al controlador que resolgui l'encaixada, l'existència de repositoris centralitzats dels objectes que cal recuperar en la resolució de l'encaixada dona un disseny amb controladors molt poc cohesionats i fortament acoblats.

¹⁷Els *metaprincipis* vistos a l'apartat 2.4.Assignació de responsabilitats, pàgina 25, → expressen com usar els principis de disseny; el *metaprincipi* de la *Solidesa argumentada* expressa com s'ha de fer i presentar el disseny.

2.4.Assignació de responsabilitats, pàgina 25

Principis i definicions del capítol

Principis

Cistella, 352
Comunicació amb
identificadors, 341
Connexions, 345
Encaixada, 352
Identificació referencial, 353
Identificadors i Presentació,
341
Solidesa argumentada, 355

Definicions

Desenvolupament per
capes (primera
aproximació), 337
Desenvolupament per capes
(proposta definitiva),
340
Especificació per capes, 339
Resolució de l'encaixada, 353
Sistema de tres capes, 338

Capítol 18

GLS Identificadors i *novaInscripció (caminada)*

1 La caminada com a identificador

1.1 Anàlisi de la proposta

Validesa de la proposta. L'ES *novaInscripció (caminada)* envia al controlador un argument, de nom *caminada*. Assumim que és un *identificador* d'una realització de *Caminada*. Aquesta assumptió està d'acord amb el principi de les *Connexions*.[→]

17.2.3.Principi de les *Connexions*, pàgina 345

1.2 Necessitats de l'encaixada

Un identificador que cal encaixar. Segons el principi de l'*Encaixada*[→] la primera responsabilitat que cal resoldre és la recuperació de la *c : Caminada* identificada per *caminada*.

17.4.Encaixada, pàgina 352

Encaixada possible. El principi de les *Connexions*[→] exigeix que la *recuperació referencial* de l'identificador donat.¹ És a dir, que el sistema té els mecanismes necessaris per a poder obtenir qualsevol *: Caminada* a partir del seu valor d'identificació. Per tant l'encaixada és possible.

17.2.3.Principi de les *Connexions*, pàgina 345

Visibilitat per a l'encaixada. Si assumim l'existència d'un *repositori centralitzat* de *Caminades*, amb visibilitat des del controlador, podem resoldre fàcilment l'encaixada: n'hi haurà prou en fer un *find()* sobre el repositori emprant l'identificador donat.

Un objecte per a la visibilitat d'encaixar. El modisme *MOMO*[→] exigeix considerar el conjunt d'objectes que pertanyen a la destinació d'un mateix

5.5.MOMO: Un modisme pels multiojectes, pàgina 100

¹La recuperació referencial és per a l'identificador. Cal tenir en compte que *caminada* és un valor d'identificació, i no la propietat d'identificació o identificador.

enllaç dirigit realització de $K \rightarrow \text{Caminada}^*$ com un objecte d'implementació que el disseny s'ha de preocupar de crear.

Creació primerenca, pàgina ??

MOMO i Creació primerenca. La *Creació primerenca*[→] demana que l'objecte exigít per MOMO es creï juntament a la creació de l'objecte origen de l'enllaç dirigit. És a dir, que en el moment de crear el controlador s'ha de crear el multiobjecte que manté les caminades.

Necesitat de l'encaixada. Per altra banda, en el moment de fer l'encaixada de la caminada² el multiobjecte exigít per MOMO ja ha d'estar poblat. És a dir, en el moment de fer l'encaixada de la caminada cal que les caminades ja estiguin en el repositori del controlador.

1.3 Una condició que facilita l'encaixada

Una condició suficient. Si fem que en crear el controlador aquest creï l'enllaç dirigit multiavaluada de les caminades (i no només l'objecte exigít per MOMO)³ assegurem que en el moment de l'encaixada de la caminada el sistema ja té les caminades.

? ! (49)

- El sistema conté un repositori centralitzat de Caminades
- En la inicialització del sistema es dona al controlador visibilitat d'atribut sobre aquest repositori

1.4 Comprovació d'identificador

Comprovació d'identificador. Com que tenim un ES amb un identificador com a argument, s'aplica el principi de la *Identificació robusta*, que exigeix l'existència d'un mecanisme de *comprovació d'identificador* sobre les Caminades:

existeixCaminada (nom)

Contracte <i>existeixCaminada (nom)</i>	
Paràmetres	
• <i>nom</i> :	Identificador de la caminada
PRE	
1. Cert	
POST	
1. Cert	\Leftrightarrow En el sistema existeix una <i>c</i> :Caminada amb <i>c.nom</i> = <i>nom</i>

²Pròpiament el que encaixem és un valor d'identificació d'un objecte :Caminada.


³El multiobjecte exigít per MOMO es pot veure com l'espai on desar les caminades. El que demanem ara és que en crear aquest espai també l'omplem.

1.5 Conclusions

Conclusions. Les conclusions queden sumaritzades en el següent conjunt de decisions presentat en forma d'alternativa.⁴

- caminada és un **identificador**
- Caminada té recuperació referencial
- Cal un ES *existeixCaminada (nom)*

Les assumpcions de 49 \rightarrow faciliten el disseny tant de la *recuperació referencial* com de la *comprovació d'identificador*.

(50) 

Assumpció 49, pàgina 358

2 La caminada com a objecte


2.1 Anàlisi de la proposta

Validesa de la proposta. L'ES *novaInscripcio (camnada)* envia al controlador un argument, de nom *camnada*. Assumim que és un *objecte* realització de *Caminada*. Aquesta assumpció està d'acord amb el principi de les *Connexions*.

Exigències de les Connexions. El principi de les *Connexions* exigeix que el sistema ofereixi en la seva interfície un mecanisme per a la recuperació exhaustiva de *Caminada*.

- caminada és un **objecte**
- Caminada té recuperació exhaustiva visible en la interfície del sistema


Decissions vàlides sense necessitat de cap assumpció.

(51) 

⁴L'altra alternativa és el cas en què considerem que *camnada* és un objecte, i que analitzem en el següent apartat.

2.2 Una possible recuperació exhaustiva

Un a possible recuperació exhaustiva. Un cas particular de *recuperació exhaustiva* és la *recuperació referencial*.

 (52)

Considerem la següent recuperació exhaustiva (en concret, referencial):

obtenirCaminada (nom) : <c:Caminada>

Contracte <i>obtenirCaminada (nom) : <c:Caminada></i>	
Paràmetres	
• <i>nom</i> :	Identificador de la caminada
PRE	
1. <i>c:Caminada</i>	
POST	
1. <i>c.nom=nom</i>	

18.1.La caminada com a identificador, pàgina
357

Un camí ja fressat. Més amunt[→] hem vist com si la caminada és un identificador cal un mecanisme de *recuperació referencial*. Ara estem analitzant el cas en què la *caminada* és un objecte, i assumim que resollem l'exigència de la *recuperació exhaustiva* demanada per les *Connexions* en termes de *recuperació referencial*.

Recuperació referencial amb la *caminada* com a identificador. Si la *caminada* és un identificador la *recuperació referencial* és obligada.[→] Però els principis no ens diuen res de si ha de ser visible a la interfície del sistema o no.

Recuperació referencial amb la *caminada* com a objecte. Si la *caminada* és un objecte la *recuperació referencial* és un dels mecanismes possibles per a resoldre la *recuperació exhaustiva*. Però, en cas d'usar-la, les *Connexions* exigeixen que ha de ser visible en la interfície del sistema, és a dir, que *obtenirCaminada (nom)* ha de ser un ES.

Comprovació d'identificador. Tenir la *recuperació referencial* visible en la interfície del sistema significa que tenim un ES amb un identificador de *Caminada* com a argument. I llavors s'aplica el principi de la *Identificació robusta*, que exigeix l'existència d'un mecanisme de *comprovació d'identificador* sobre les *Caminades*:

existeixCaminada (nom)

Contracte <i>existeixCaminada (nom)</i>	
Paràmetres	
• <i>nom</i> :	Identificador de la caminada
PRE	
1. Cert	
POST	
1. Cert \Leftrightarrow	En el sistema existeix una <i>c</i> :Caminada amb <i>c.nom</i> = <i>nom</i>

2.2.1 Una condició que facilita la recuperació referencial

Res de nou. L'encaixada és una recuperació referencial. A 18.1.La caminada com a identificador, pàgina 357,[→] hem vist com l'existència d'un repositori centralitzat facilita aquesta recuperació referencial. Ara tenim la mateixa situació; l'única diferència és que exigim que la recuperació referencial sigui un ES.


18.1.La caminada com a identificador, pàgina 357

2.3 Conclusions

Conclusions. Les conclusions queden sumaritzades en el següent conjunt de decisions presentat en forma d'alternativa.⁵

- caminada és un **objecte**
- Caminada té recuperació exhaustiva visible en la interfície del sistema
 - Si aquesta recuperació la tenim en forma de *recuperació referencial* llavors:
 - * Cal un ES *existeixCaminada (nom)*

Les assumpcions de 49 [→] faciliten el disseny tant de la *recuperació referencial* com de la *comprovació d'identificador*.

(53) 

Assumpció 49, pàgina 358

⁵L'altra alternativa és el cas en què considerem que *caminada* és un objecte, i que analitzem en el següent apartat.

Capítol 19

Àmbits d'identificació

1	Identificació i espai d'identificació	366
2	Els identificadors en el MC	367
2.1	El concepte com a espai d'identificació	367
2.2	Conceptes febles	368
2.3	Els conceptes febles en el MC	372
3	Els identificadors en el MComp	374
3.1	El paper dels identificadors a <i>MComp</i>	374
3.2	Emmagatzematge	375
4	Identificadors i repositoris	375
4.1	Àmbit d'identificació	375
4.2	Selecció de l'àmbit d'identificació	376
4.3	Objectes i repositoris	378
5	Identificació atributiva	379
6	Visibilitat dels àmbits d'identificació	381
	Principis	383
	Definicions	383

Contingut detallat del capítol 19

1 Identificació i espai d'identificació	366
Punt de partida	366
El component com a espai d'identificació	366
Llenguatge a dos nivells	367
Generalització del llenguatge a dos nivells	367
Un pas més enllà	367
2 Els identificadors en el MC	367
2.1 El concepte com a espai d'identificació	367
Identificació en el MC	367
Identificació d'enllaços del MC	368
Identificació de conceptes associatius en el MC	368
És el concepte un context d'identificació necessari	368
2.2 Conceptes febles	368
2.2.1 Definició i exemples	368
El perquè presentem els conceptes febles	368
2.2.2 Comunicació dels conceptes febles	371
Llenguatge a dos nivells	371
Quants nivells?	371
Concepte fort que és feble	371
Comunicació per nivells	371
Aspectes fonamentals de la comunicació per nivells	371
2.3 Els conceptes febles en el MC	372
Pregunta mig resposta	372
Preguntes pendents	372
Full de ruta	373
Representació en Chen	373
Representació en UML	373
Excursió	373
Context.	373
Proposta de representació.	373
El model en termes descriptius.	373
Model forçat.	373
Múltiples valors d'identificació.	373
Motius d'una absència	373
Delegació en el disseny de la identificació	374
3 Els identificadors en el MComp	374
3.1 El paper dels identificadors a <i>MComp</i>	374
El component com a espai d'identificació	374
Comunicació amb l'exterior	374
Encaixada	374
Identificadors i multiobjectes	374
Realitzacions actives	374
Els identificadors en el disseny	375

3.2 Emmagatzematge	375
Multiobjectes i recuperació referencial	375
Un multiobjecte és un repositori	375
Repositori fragmentat	375
4 Identificadors i repositoris	375
4.1 Àmbit d'identificació	375
Unicitat de la denominació	375
Relaxació de la unicitat	375
El component com a espai d'identificació	375
Relaxació en el repositori	376
Espai i àmbit d'identificació	376
Identificació global i local	376
4.2 Selecció de l'àmbit d'identificació	376
Definició d' <i>àmbit d'identificació</i>	376
Àmbit i propietat d'identificació	376
Consistència individualitzadora	377
Identificació interna	377
Identificació externa	377
4.3 Objectes i repositoris	378
Excursió. (Duplicació de realitzacions)	378
Excursió	378
Excursió. (Rèpliques i compartició)	379
Rèpliques i fragments	379
5 Identificació atributiva	379
La propietat d'individualització	379
La individualització com a propietat d'associació	379
Variabilitat identificadora	379
Traçabilitat	380
Estabilitat d'identificació	380
<i>Estabilitat d'identificació</i> i disseny	380
Immutabilitat de l'identificador	380
Propietat d'identificació i individualització	380
Identificació atributiva	380
Propietat d'identificació i identificació atributiva	380
6 Visibilitat dels àmbits d'identificació	381
Responsabilitats en la identificació	381
Visibilitats del mecanisme de <i>comprovació d'identificador</i>	381
Àmbits accessibles	381
Àmbit accessibles i identificació referencial	381
Àmbits accessibles i comprovació d'identificació	381
Principis	383
Definicions	383

1 Identificació i espai d'identificació

15. Identificadors, pàgina 277

Punt de partida. A 15. Identificadors, pàgina 277[→], s'han introduït els formalismes sobre els *identificadors*. De tots aquests formalismes les idees que ens interessen, expressades en termes informals, són:

Definició pàgina 289. (15.3.2. Identificació i propietat d'identificació)

- **Propietat d'identificació.** [→] És una propietat d'un conjunt de realitzacions, anomenat *espai d'identificació*, tal que:

1. *Obligatorietat.* Cadascuna de les realitzacions de l'espai d'identificació ha de tenir un valor per a aquesta propietat
2. *Unicitat.* No hi pot haver dues realitzacions diferents dins l'espai d'identificació amb el mateix valor per a aquesta propietat

- **Valor d'identificació.** És el valor que una realització concreta té per a la propietat d'identificació indicada. El *valor d'identificació* permet *individualitzar*, dins l'espai d'identificació, la realització de les altres realitzacions

- **Identificador.** El context ha de permetre desambiguar si estem parlant d'una *propietat d'identificació* o d'un *valor d'identificació*.

Definició pàgina 294. (15.3.5. Propietat d'identificació especular)

- **Identificació especular.** [→] Totes les propietats d'identificació que considerem són aquelles que el seu espai d'identificació és un component, i tals que els seus valors tant permeten individualitzar una realització de la realitat, com la corresponent realització del model.

15.2.3. Múltiples llenguatges, pàgina 285

- **Llenguatge a dos nivells.** [→] Tota abstracció té un nom que li és propi. Per expressar una identificació cal indicar el nom de l'abstracció i el valor d'identificació dins d'aquesta abstracció. Així el nom d'una realització és sempre un parell `<nom abstracció, identificador >`

(76)

El paper dels identificadors.

Els identificadors són un mecanisme *individualitzador* de la col·lectivitat representada per l'*espai d'identificació*.

15.4. Component ben definit, pàgina 296

El component com a espai d'identificació. A 15.4. Component ben definit, pàgina 296[→], hem definit formalment com han de ser els components per tal que el model sigui comunicable. En termes planers demanàvem:

- *Comunicable* El component té un nom únic en el model. Aquest nom coincideix amb exactament un concepte de la realitat.
- *Estructura comunicable.* Els objectes realització del component es corresponen amb entitats que són realitzacions del concepte modelitzat per aquest component, i a la inversa.

19.2 Els identificadors en el MC

J.M. Merenciano

- *Espai d'identificació el component.* El component és un espai d'identificació. És a dir, tenim una *propietat d'identificació* que impedeix que dues realitzacions del mateix component tinguin el mateix valor per a aquesta propietat.¹
- *Identificació especular.* Un mateix valor d'identificació individualitza una realització en el context del component, i en el context del concepte que aquest modelitza

Llenguatge a dos nivells. En la comunicació fem el *llenguatge a dos nivells*. → En la presentació i llenguatge per nivells diem que cada realització es comunica juntament amb el nom de l'abstracció de la qual n'és realització.

Conclusió 53, pàgina 286. (15.2. Llenguatge per nivells)

Generalització del llenguatge a dos nivells. Ara podem expressar el llenguatge a dos nivells com aquell en el que per a la comunicació d'una realització cal comunicar el seu valor d'identificació, i l'espai d'identificador on aquest valor és vàlid: `<espaiIdent, identificador>`.

En el *llenguatge a dos nivells* cada realització es comunica juntament l'espai d'identificació → on s'ha d'interpretar el nom comunicat.

(77)

Definició pàgina 290. (15.3.2. Identificació i propietat d'identificació)

Un pas més enllà. Hem vist → com si l'espai d'identificació que es comunica és sempre el d'un component, la comunicació és possible. Però és l'única possibilitat?

15.5. Identificació i llenguatge de comunicació, pàgina 298

És el component/concepte l'únic espai d'identificació possible de cara a la comunicació?²

(54) ?

2 Els identificadors en el MC

2.1 El concepte com a espai d'identificació

Identificació en el MC. En el MC es representen els conceptes i les associacions entre aquests. Els identificadors sobre els conceptes permeten individualitzar les seves realitzacions; però també els enllaços i les realitzacions de conceptes associatius.¹ Tot seguit ho repassem.

¹Aquesta és la condició d'*unicitat* de la propietat d'identificació. A més tenim la condició d'*obligatorietat*.

²Parlar del component com a context d'identificació és també parlar del concepte com a context d'identificació. Això és conseqüència de la condició d'*identificació especular* que exigim als components del nostre model.

(78)

Per a individualitzar qualsevol realització de la realitat és suficient en què tot *espai d'identificació* sigui un *concepte*

Identificació d'enllaços del MC. Donada una associació n -ària, amb un extrem de l'associació amb multiplicitat 1, per identificar un enllaç cal identificar les realitzacions dels $n - 1$ extrems de l'enllaç que no es corresponen a l'extrem amb multiplicitat 1 de l'associació;³ si tots els extrems de l'associació n -ària tenen multiplicitat diferent de 1, llavors l'enllaç s'identifica amb la identificació de les realitzacions dels seus n extrems.

Identificació de conceptes associatius en el MC. Per identificar la realització d'un concepte associatiu cal identificar l'enllaç amb qui s'enllaça.

És el concepte un context d'identificació necessari. Sabem que si usem el *concepte* com e l'*espai d'identificació*, totes les realitzacions de la realitat es poden individualitzar, i la comunicació és possible. Però cal usar el concepte com a context o hi ha altres possibilitats?

2.2 Conceptes febles

2.2.1 Definició i exemples

El perquè presentem els conceptes febles. El *concepte feble* és un element important del model conceptual, però no és imprescindible.⁴ Aquí el plantejarem per veure com a nivell d'especificació hi ha d'altres espais d'identificació possibles, a part dels conceptes.

! (55)

La definició formal d'un *concepte feble* és complexa, i la seva comprensió no és necessària per a seguir la lectura. Per tant la lectura i comprensió dels apartats d'aquest capítol que parlen del conceptes febles es pot ometre totalment.

Definició. Concepte feble

Sigui una associació $f:A-B$, obligada per l'extrem A . Direm que B és un concepte feble respecte de A i f si i només si existeix un subconjunt X de A , $X \in A$, tal que per cada $x \in X$ hi ha una propietat d'identificació sobre $f(x)$:

$$(\forall x \in X \subseteq A)(\exists Prop_{Id}(f(x)))$$

³Si hi ha més d'un extrem de l'associació amb multiplicitat 1, continua sent vàlida la necessitat d'identificar els objectes de $n - 1$ extrems.

⁴L'UML ni tant sols el considera.

Definició. Concepte fort

Donat un concepte feble B sobre A i f direm que A és el concepte fort corresponent a B segons f ,

Definició. Associació d'identificació

Donat un concepte feble B sobre A i f direm que f n'és l'associació d'identificació

Concepte feble en termes planers.

Un concepte feble B és aquell que:

- *Dependència.* Només té sentit en relació a un altre concepte A , que anomenem *fort*.⁵
- *Identificació a dos nivells* La identificació d'una realització del concepte feble significa identificar la realització pertinent del concepte fort, més la identificació dins l'espai d'identificació definit per la realització del concepte fort:

$\langle \text{ident de } x:A, \text{ident dins } f(x) \rangle$

(79)

Exemple 82 (Concepte feble) En els equipaments de convencions s'acostuma a donar noms a les sales en funció dels seus usos o en homenatge a personatges.

Cada centre de convencions, però, és totalment sobirà a l'hora de posar els noms a les seves sales. I per tant és molt possible que tant el Palau de congressos de Barcelona, com l'Auditori de Vilanova tinguin una sala anomenada "Sala d'actes" o bé una sala "Bernat Metge". El nom de la sala, per tant és ambigu, no hi ha unicitat.

Però si sabem en quin centre de convencions estem llavors l'ambigüitat desapareix i obtenim la unicitat.

En d'altres paraules, la propietat NomSala no és suficient per a individualitzar una sala; no és una propietat d'identificació de Sala.

Per poder individualitzar una sala cal indicar el valor de la propietat NomSala d'aquesta juntament amb el nom del centre de convencions que estem considerant. El motiu és que cada centre de convencions defineix un espai d'identificació de les sales. Només si coneixem a quin centre ens estem referint podrem individualitzar la sala "Bernat Metge".

⁵La definició exigeix que tot $b:B$ formi part d'algun espai d'identificació. I aquests espais d'identificació els defineixen un subconjunt d'entitats de A . Per tant, no podem donar nom a un $b:B$ si no és dins de l'espai d'identificació definit per un $x:A$.

En el cas que ens ocupa, doncs, per individualitzar una sala cal indicar el nom del centre de convencions, i el nom de la sala. Per tant, la propietat d'individualització d'una sala és $Prop_{Id}(\text{Centre}) \times Prop_{Id}(\text{sales}(\text{centre}))$.⁶

Conclusió. L'espai d'identificació del concepte feble és una entitat del concepte fort corresponent.⁷

El l'espai d'identificació de *Sala* és un :*Centre* concret.

Exemple 83 *Ordre com a concepte feble* Un cas molt habitual, que es pot representar en termes de concepte feble, és el de les associacions amb ordre.

Per exemple, en un diari de bitàcola s'hi fan múltiples anotacions ordenades. Per tant tenim una associació registre entre Bitàcola i Anotació, obligada per l'extrem de la Bitàcola.

Una manera d'expressar això en el MC és amb un estereotip sobre l'extrem Anotació de l'associació, que digui que els enllaços estan ordenats.

Una altra manera és a través dels conceptes febles. Anotació és un concepte feble respecte Bitàcola i registre. Donada una *b:Bitàcola* qualsevol aquests defineix un espai d'identificació compost per totes les anotacions enregistrades en aquesta bitàcola. Sobre aquest espai d'identificació podem definir una propietat d'identificació que no és més que un comptador.

Amb els conceptes febles té sentit parlar de l'anotació 3 de la bitàcola d'en Joan, i de l'anotació 3 de la bitàcola de la Remei.

Exemple 84 (Unicitat i coneptes febles) *La Sala és un concepte feble de Centre. Com a identificador de Sala considerem el seu nom.*

Això significa que hi poden haver sales amb el mateix nom, sempre i quan siguin de centres diferents.

! (56)

Donat un concepte feble és perfectament possible que dues realitzacions tinguin el mateix valor per a la propietat d'identificació. Això és així perquè la propietat d'identificació només es comporta com a tal (amb obligatorietat i amb unicitat) dins l'espai d'identificació; i aquest el defineix una realització del concepte fort.

⁶La primera propietat d'identificació té com a espai d'identificació el concepte *Centre*, i per tant ens permet identificar un centre qualsevol. Un cop el centre identificat ja tenim la realització del concepte fort que defineix l'espai d'identificació del concepte feble. En concret, assumim una associació *sales* que donat un centre ens dona les severs sales. Llavors l'espai d'identificació de la sala és *sales(centre)*, on *centre* és la realització identificada del concepte fort.

⁷Per ser precisos, l'espai d'identificació és el conjunt d'entitats associades a una realització del concepte fort en virtut de l'associació d'identificació. Sovint però considerarem que l'associació d'identificació és deduïble del model (entre el concepte fort i el feble només hi ha una associació, i aquesta és justament l'associació d'identificació), i per tant no l'explicitarem.

2.2.2 Comunicació dels conceptes febles

Llenguatge a dos nivells. L'espai d'identificació del nom d'un concepte feble és el conjunt d'entitats enllaçades, segons l'associació d'identificació, a una realització del concepte fort. Per tant en la comunicació haurem de comunicar:

- Espai d'identificació:
 - Associació d'identificació
És el nom d'una abstracció
 - Una realització del concepte fort
Per comunicar-la caldrà comunicar el seu nom (identificador)⁸ i l'espai d'identificació d'aquest nom (és a dir, el nom del concepte fort)
- Nom de la realització del concepte feble
Valor d'identificador definit sobre l'espai d'identificació expressat pel context interpretatiu comunicat

Quants nivells? En la formulació del llenguatge per nivells dèiem que comunicàvem el valor d'identificació de la realització, i el seu espai d'identificació (que en aquell moment era el nom d'un component). Ara veiem que l'espai d'identificació pot estar definit per una realització d'un altre component. Per tant el context té dos nivells:⁹

$\langle \langle \text{nom component fort, id fort} \rangle, \text{id feble} \rangle$

Concepte fort que és feble. Suposem que tenim un concepte feble C respecte B . Llavors B és fort respecte a C ; però res impedeix que B sigui feble respecte A . Així, en el paràgraf anterior caldrà substituir en el tuple l'expressió "component fort" per "un parell format pel nom del component fort A i el nom d'una realització de A' ". El resultat és que el nombre de nivells del llenguatge augmenta.

L'aparició dels conceptes febles converteix el llenguatge a dos nivells en un llenguatge de n nivells

(80)

Comunicació per nivells. La presència dels *conceptes febles* fa que el llenguatge de comunicació passi de dos a un nombre indeterminat de nivells. Tot seguit resumim les idees claus, sense fer-ne una definició precisa. L'objectiu no és aprofundir en els *conceptes febles* sinó mostrar com l'ús dels conceptes com a espai d'identificació no és pas l'única possibilitat.

⁸Si el concepte fort és un altre concepte feble o bé és un concepte associatiu, llavors el nom de la realització no és un nom, ans una seqüència de noms. Per exemple, en el cas d'un concepte associatiu, caldrà indicar el nom del concepte associatiu, i el nom de les dues realitzacions enllaçades.

⁹Per simplicitat no explicitem l'associació d'identificació.

Aspectes fonamentals de la comunicació per nivells. L'estudi (superficial) que hem fet als *conceptes febles* nostra que:

- *Comunicació de l'espai d'identificació.* En la comunicació sempre cal comunicar també l'espai d'identificació de l'identificador passat
- *Hi ha alternatives.* Els espais d'identificació no necessàriament es limiten als conceptes; poden ser un subconjunt d'aquests
- (En el cas dels *Conceptes febles* l'espai d'identificació és un subconjunt del concepte feble; i es defineix a partir de l'associació d'identificació i d'una entitat del concepte fort)

(81)

Llenguatge de comunicació per nivells.

En el llenguatge a múltiples nivells cada realització es comunica juntament amb el context on s'ha d'interpretar el nom comunicat (el seu espai d'identificació)

(82)

Possibles espais d'identificació.

- Un component

L'espai d'identificació és el conjunt de totes les entitats que són realització del concepte donat

- Un subconjunt d'un component, definit per una entitat¹⁰ i una associació

L'espai d'identificació és el conjunt d'entitats associades a l'entitat donada en virtut de l'associació indicada

2.3 Els conceptes febles en el MC

Pregunta mig resposta. La pregunta 54, pàgina 367[→], planteja si el *concepte* és l'únic espai d'identificació possible. Part de la resposta l'acabem de veure: els *conceptes febles* impliquen espais d'identificació que són un subconjunt del concepte

¹⁰Si l'espai d'identificació es defineix a partir d'una entitat, caldrà comunicar aquesta; i això significa comunicar el seu identificador i el seu espai d'identificació. El nombre de nivells resultants és indeterminat. Per exemple, en el cas de tenir un concepte feble tal que el seu concepte fort és un concepte associatiu, per exemple, llavors l'espai d'identificació es defineix a partir d'una entitat associativa. Per comunicar-la caldrà indicar les entitats enllaçades, el nom del concepte associatiu, i el nom de l'associació d'identificació.

Preguntes pendents. Queden pendents encara algunes preguntes:

- Què passa però a MComp?[→]
- Quin és el model d'un *concepte feble*[→]
- Com es representen al MC els conceptes febles?

???, pàgina ??

???, pàgina ??

Full de ruta. Començarem per la darrera d'aquestes preguntes. La segona de les preguntes serà tractada en el següent apartat, i la pregunta restant la deixem per un altre capítol.

Representació en Chen. En el llenguatge de modelització *Chen* o *E-R* els conceptes febles són ciutadans de primera categoria: el concepte feble es representa amb un doble emmarcat; i l'associació d'identificació amb doble ratlla.

Representació en UML. En UML els conceptes febles no estan representats. I pitjor encara, qualsevol intent de representar-los no és prou satisfactori.

Excursió Conceptes febles en UML.

Context. Sigui B un concepte feble respecte A i f ; i sigui $nom \in Prop_{Id}(f(a))$, per a un $a : A$ qualsevol.

Proposta de representació. Una possible representació és:

- Introduïm un concepte *Nom* que contingui els noms donats a totes les realitzacions de B ; és a dir, la unió de l'abast de totes les propietats d'identificació considerades:

$$\bigcup_{a:A} abast(Prop_{Id}(f(a)))$$

- Convertir B en un concepte associatiu entre A i Nom

El model en termes descriptius. La idea és que tenim un concepte amb tots els noms. Donat un $:Nom$ i una realització del concepte fort només hi ha una realització possible pel concepte feble. I donada una realització del concepte feble aquesta té un nom interpretable en el context d'una única realització del concepte fort. D'aquí que el concepte feble aparegui com a concepte associatiu.

Model forçat. El model proposat exigeix un concepte amb tots els noms. És a dir, un concepte que només té els vlors d'identificació que admetrem en els diferents espais d'identificació. Semànticament, pel nostre problema, no té cap importància; és una necessitat de l'UML. El resultat és un model forçat, artificios i farragós.

Múltiples valors d'identificació. Un altre "però" d'aquesta proposta és que no admet múltiples valors d'identificació per a una mateixa realització: el concepte associatiu assegura que cada realització del concepte feble és interpretable en l'espai d'una única realització del concepte fort, i dins d'aquest espai té un únic nom.

Motius d'una absència. Perquè l'UML no contempla els conceptes febles? Aquesta pregunta pot tenir més d'una resposta, però potser una de les més clares és la *delegació en el disseny*.¹¹

(83)

L'UML delega en el disseny qualsevol decisió sobre els identificadors

??,?, pàgina ??

Delegació en el disseny de la identificació. Com veurem[→] en el disseny disposem d'eines que permeten expressar fàcilment les condicions que cal posar als valors d'identificació dels models dels conceptes febles. @@@@

3 Els identificadors en el MComp

3.1 El paper dels identificadors a MComp

El component com a espai d'identificació. En el *MComp* els components són el model dels conceptes, i les visibilitats d'atribut el model de les associacions. A 15.5. Identificació i llenguatge de comunicació, pàgina 298, [→] hem vist que la propietat d'identificació d'una associació també ho és de la visibilitat que la modelitza. Com a conseqüència els identificadors sobre els components permeten individualitzar les seves realitzacions, però també els enllaços dirigits i les realitzacions del model d'un concepte associatiu.

15.5. Identificació i llenguatge de comunicació, pàgina 298

(84)

Per a individualitzar qualsevol realització del model és suficient en què tot *espai d'identificació* sigui un *component*

17.3.5. Principi de la *Cistella*, pàgina 352

17.2.3. Principi de les *Connexions*, pàgina 345

Comunicació amb l'exterior. El principi de la *Cistella*[→] diu que el sistema internament no manipula identificadors; el principi de les *Connexions*[→] ens diu que els identificadors poden aparèixer com a arguments dels ES per tal de permetre la comunicació entre dos mons diferents. En conseqüència *els identificadors s'han de veure com un mecanisme de comunicació amb l'exterior, i no com un mecanisme intern del sistema*.

Encaixada. El principi de l'*Encaixada* diu que tant aviat com sigui possible els identificadors rebuts com a arguments d'un ES s'han d'usar per recuperar l'objecte identificat. En d'altres paraules, que la *recuperació referencial* s'ha de fer com abans millor.

Identificadors i multiobjectes. L'*identificador* és el valor que usa el mètode `find()` per recuperar una realització de dins d'un multiobjecte. És a dir, el `find()` és el mètode de recuperació referencial dins d'un repositori d'un nivell. @@No està definit encara

¹¹Aquesta explicació és només una hipòtesi, però hi ha força arguments a favor d'ella.

Realitzacions actives. Les realitzacions actives són aquelles sobre les que el controlador en té visibilitat generalitzada d'atribut, mitjançant l'anomenada *visibilitat d'activació*.[→] Per tant no cal cap identificador per fer-ne la recuperació.

Els identificadors en el disseny. De totes les observacions anteriors podem concloure que els identificadors en el disseny són un element temporal (de cara a la comunicació amb l'exterior). L'única finalitat de l'identificador és poder fer-ne la *recuperació referencial*, que per altra banda ha de ser com abans millor.

12.Arguments implícits, pàgina 217

3.2 Emmagatzematge

Aquest apartat s'ha de refer un cop escrit el capítol dels repositoris. El contingut és correcte, però segurament caldrà fer més explicacions i pot ser que canviï lleugerament la nomenclatura

(57) !

Multiobjectes i recuperació referencial. Un multiobjecte és un espai d'emmagatzematge. Les operacions sobre els multiobjecte permeten l'emmagatzematge de noves realitzacions (`add()`) i la recuperació referencial (`find()`).

Un multiobjecte és un repositori. Recordatori. Un repositori d'un sol nivell és un multiobjecte.

Repositori fragmentat. Un repositori fragmentat és un conjunt de repositoris germans. Si A fragmenta B direm que A emmagatzema B. Però el que realment volem dir és que cadascuna de les realitzacions de A emmagatzema té visibilitat d'un multiobjecte que emmagatzema un subconjunt de realitzacions de B,

4 Identificadors i repositoris

4.1 Àmbit d'identificació

Unicitat de la denominació. Per tal que els noms puguin servir com a mecanisme de comunicació cal que siguin no ambigus; és a dir, cal que cada nom expressi un sol element.[→]

14.4.3.2.Ambigüitat denominadora, pàgina 265

Relaxació de la unicitat. En la comunicació usem el llenguatge per nivells. Això permet relaxar la condició d'unicitat dins de l'espai d'identificació: n'hi ha prou en demanar que dues realitzacions diferents dins del mateix espai d'identificació no tinguin el mateix nom. La unicitat es manté ja que el nom comunicat és el parell `<espai, identificador>`.

El component com a espai d'identificació. En el model considerem com a fragment constitutiu el component. La *propietat d'identificació* sobre un component és el mecanisme emprat per anomenar les seves realitzacions. És a dir, l' *espai d'identificació* és el conjunt format per totes les realitzacions d'un mateix component.

Relaxació en el repositori. La condició de la unicitat en la denominació encara es pot relaxar més en el model: en concret n'hi ha prou en demanar la unicitat només dins d'un repositori. D'aquesta manera objectes que són realitzacions diferents d'un mateix component poden tenir el mateix identificador sempre i quan s'emmagatzemin en repositoris diferents.

Definició. Àmbit d'identificació

Anomenem àmbit d'identificació d'un component C al repositori de C en el qual el valor de l'identificador ha de ser únic.

Espai i àmbit d'identificació. L' *espai d'identificació* és el conjunt de realitzacions on hem definit una propietat d'identificació. En la definició només hi entren les condicions d'obligatorietat i unicitat. L' *àmbit d'identificació* hi afegeix la condició d'emmagatzematge.

(85)

Espai i àmbit d'identificació.

L' espai d'identificació és un terme tant de l'especificació com del disseny. L' *àmbit d'identificació* és un terme del disseny.

Identificació global i local. En el cas que l'àmbit d'identificació és un repositori centralitzat, direm que l'identificador és d' *àmbit global* . Si l'àmbit d'identificació és un repositori distribuït direm que l'identificador és d' *àmbit local* .

4.2 Selecció de l'àmbit d'identificació

Definició d'àmbit d'identificació. L' *àmbit d'identificació* s'ha definit com el repositori on s'ha de produir la individualització. Es tracta per tant d'un concepte propi del disseny, que no té lloc en el model conceptual.

Àmbit i propietat d'identificació. L' *àmbit d'identificació* s'ha introduït com un mecanisme de disseny per relaxar la condició d'unicitat de la *propietat d'identificació* : \rightarrow l' *àmbit d'identificació* és el "fragment" on els noms constitueixen una *propietat d'identificació* .

? (58)

Davant diferents alternatives quin és l'àmbit d'identificació correcte?



Consistència individualitzadora. L'àmbit d'identificació ha de ser **consistent** amb el MC.

La consistència significa que l'àmbit d'identificació del component C és el de l'objecte $i : I$ llavors l'explosió^{12,→} $i : I \Rightarrow^* C^*$ està formada per visibilitats que no violen *Espill*.

??,?, pàgina ??

@@@@@@@@@@@@@@@@@ PARèntesi

Identificació interna. En el cas de la *identificació interna*,^{13,→} l'àmbit d'identificació queda definit per les visibilitats de l'*individualitzador*. El principi de la *Individualització efectiva* diu que l'*individualitzador* és el *creador* de l'objectiu. El principi del *Creador canònic* ens diu que aquestes visibilitats han de ser consistents amb el MC. Per tant el principi de la *Consistència individualitzadora* no ens diu res de nou; és el resultat d'aplicar d'altres principis coneguts.

16.2. Identificació interna i externa, pàgina 326

Identificació externa. En el cas de la *identificació externa*, l'àmbit d'identificació és el repositori sobre el que actua el mecanisme de *comprovació d'identificador*.¹⁴ El responsable de mantenir aquest repositori és un clar candidat a *creador orb*: emmagatzema i en ser visible des del controlador aquest li pot delegar l'expertes necessària per a la creació. El principi de la *Consistència individualitzadora* demana que sigui també un candidat a *creador canònic*. La motivació d'aquesta exigència és la mateixa que la del principi del *Creador canònic*: la disminució dels acoblaments.

@@@@@@@@@@@@@@@@@ Fi parèntesi

@@@@@@@@@@@@@@@@@ Capítol GLS

Exemple 85 (Diferents àmbits d'identificació)

- *ES: novaCaminada (nom). Identificació externa. PRE: No existeix cap Caminada amb aquest nom. Àmbit d'identificació global.*

La identificació externa obliga la PRE: cal assegurar que l'argument és un identificador vàlid. La comprovació de l'identificador exigeix que el controlador tingui visibilitat de totes les Camidades.

- *ES: novaInscripció(id). Identificació externa. PRE: Tenim una c:Caminada activa; id no existeix*

¹²Recordem que quan diem que $i : I$ és un àmbit d'identificació de C estem dient que $i : I$ és un repositori de C . I això és un abreujament per dir que $i : I$ és l'origen d'una expansió, formada per visibilitats d'atribut, sobre C .

¹³La distinció entre identificació interna i externa s'ha introduït a l'apartat 16.2. Identificació interna i externa, pàgina 326

¹⁴Si veiem la comprovació d'identificador com una comprovació d'estructura i una comprovació d'existència, llavors l'àmbit d'identificació és el repositori sobre el que es fa la comprovació d'existència.

dins la c :Caminada activa. Àmbit d'identificació: local a la c :Caminada.

La identificació externa obliga la segona PRE: cal assegurar que l'argument és un identificador vàlid. La comprovació de l'identificador exigeix que el controlador tingui visibilitat (directa o indirecta) de les Inscripcions de c . El principi de la Consistència identificadora exigeix que en el MC hi hagi una associació entre Inscripció i Caminada.

- ES: $\text{novaInscripció}(c:\text{Caminada}, id)$. Aquest cas és idèntic a l'anterior, però amb l'argument c :Caminada explícit enlloc d'implícit.

4.3 Objectes i repositoris

Definició. Realització compartida

Donada una realització d'un component direm que és una realització compartida si pertany a més d'un repositori.

Excursió. (Duplicació de realitzacions) La realització compartida significa una *compartició lògica*: una mateixa realització és accessible des de dos repositoris diferents.

Un altre concepte és el de la *duplicació de realitzacions* en el que diferents còpies d'una mateixa realització s'emmagatzemen en repositoris diferents. La *duplicació de realitzacions* s'usa, per exemple, quan els diferents repositoris on volem que aparegui una mateixa realització lògica estan en màquines diferents.

En el que segueix no contemplem mai la possibilitat de duplicar una realització.

Definició. Migració de realitzacions

Anomenem migració de realitzacions al fet que una realització pugui canviar de repositori.

Excursió Migració via duplicació. Una possible manera de veure la migració d'una realització $c:C$ d'un repositori $R1$ a un repositori $R2$ és:

1. Creem un duplicat $c':C$ de $c:C$
2. Destruïm $c:C$ (i per tant desapareix de $R1$)
3. Emmagatzemem $c':C$ a $R2$

Aquesta visió implica la duplicació de realitzacions, i per tant no la considerarem.

Definició. Rèplica d'un repositori

Si el disseny assegura que tota realització $c : C$ que s'emmagatzema (ara, en el passat, o en el futur) en un repositori $R1$ també s'emmagatzema en un altre repositori $R2$, direm que $R1$ i $R2$ són rèpliques una de l'altra.

Excursió. (Rèpliques i compartició) Com que no admetem duplicacions, les rèpliques comparteixen lògicament les seves realitzacions.

Definició. Repositori replicat

Un repositori replicat és un repositori que té rèpliques.

Rèpliques i fragments. Cada rèplica pot constar dels fragments que li convinguin. Un cas molt habitual és tenir una rèplica centralitzada (un sol fragment), i una altra rèplica distribuïda en diversos fragments.

Exemple 86 De moment hem decidit que cada $c : \text{Caminada}$ emmagatzema les seves *Inscripcions*. Imaginem ara que ens veiem en la necessitat d'accedir a una $i : \text{Inscripció}$ de la que desconeixem a quin $c : \text{Caminada}$ es correspon. L'existència d'una rèplica centralitzada d'*Inscripcions*, juntament amb una identificació d'àmbit global, permet resoldre el problema. Com a resultat cada $i : \text{Inscripció}$ pertany a dos repositoris: al local de la $c : \text{Caminada}$ pertinent; i al global.

5 Identificació atributiva

La propietat d'individualització. Per saber com o qui ha de crear un identificador cal analitzar què es necessita per crear-lo. I per fer-ho hem de començar per analitzar la funcionalitat bàsica dels identificadors: la individualització.¹⁵

Què és i què significa la individualització d'un element d'un conjunt?

(59) ?

La individualització com a propietat d'associació. La individualització d'un element dins d'un conjunt no és una propietat ni de l'element ni del conjunt, sinó de la relació de pertinença de l'element dins del conjunt. És a dir, és una propietat de l'associació *pertinença* entre *Element* i *Conjunt*.

¹⁵Un identificador o valor de la propietat de la identificació és un valor individualitzador; però a més cal que tot element del col·lectiu tingui valor per a aquesta propietat. *Unicitat* i *obligatorietat* són els mots clau. Aquí analitzem la *unicitat*.

Variabilitat identificadora. El fet que la individualització és una propietat de la relació de pertinença significa que, en el disseny, l'identificador canvia davant de la migració i de la compartició.¹⁶ En el primer cas el valor de l'identificador canvia en el temps; en el segons cas una mateixa realització té diferents identificadors segons des d'on hi accedim.

Traçabilitat. Com a conseqüència la traçabilitat de les realitzacions a través dels seus identificadors es fa extremadament difícil. D'aquí el següent principi:

Principi



Estabilitat d'identificació. El valor de l'identificador, un cop assignat a una realització, no canvia: ni en el temps ni segons el repositori d'accés considerat.

Estabilitat d'identificació i disseny. El principi de l'*Estabilitat d'identificació* parla de la identificació en el disseny, i no afecta en absolut la identificació del model conceptual.

Immutabilitat de l'identificador. El principi de l'*Estabilitat d'identificació* permet que la individualització es pugui considerar una propietat de la realització, enlloc d'una propietat de la relació de pertinença. En la creació del valor de l'identificador potser entrarà en joc el conjunt; però un cop assignat un valor a l'identificador, aquest esdevé una propietat de l'element.

Propietat d'identificació i individualització. El principi de l'*Estabilitat d'identificació* diu que el valor la propietat d'identificació d'un objecte és el valor d'una propietat d'aquest objecte. I per tant que la *propietat d'identificació* és una propietat del *component*.

Principi



Identificació atributiva. La identificació és una propietat de la realització, i per tant es pot modelitzar com un atribut del component pertinent. És més, el modelitzarem com a atribut.

Propietat d'identificació i identificació atributiva. Per a tenir un *identificador*, la *Identificació atributiva* per si sola no és suficient; ha d'anar acompanyada de:

- L'*Estabilitat d'identificació*: l'identificador és un atribut no modificable
- L'*obligatorietat de l'atribut*: tot objecte té un valor per a l'atribut
- L'*unicitat*: no hi ha dos objectes amb el mateix valor dins del mateix àmbit d'identificació

¹⁶La migració i la compartició es defineixen en termes de repositoris, és a dir, en termes d'emmagatzematge. Però l'emmagatzematge només té sentit en el model de la solució (en el disseny), no pas en el model conceptual.

6 Visibilitat dels àmbits d'identificació

Responsabilitats en la identificació. El principi de la *Identificació robusta*[→] exigeix que el sistema sempre ha de tenir la responsabilitat d'oferir els mecanismes de *comprovació dels identificadors*. En el cas de la *identificació interna* el sistema també ha de tenir la responsabilitat de la *creació dels identificadors*.

16.3.3. Identificació robusta, pàgina 329

Responsabilitats del disseny envers la identificació.

(86)

El disseny és responsable de:

- Disposar dels mecanismes de **recuperació referencial** exigits per la *Identificació referencial* [→]
- Oferir a la interfície del sistema els mecanismes de **comprovació d'identificador** exigits per la *Identificació robusta*[→]
- **Crear els valors d'identificació** (només en el cas d'*identificació interna*)[→]

Principi pàgina 353, (17.4.2. Resoldre l'encaixada)

16.3.3. Identificació robusta, pàgina 329

???, pàgina ??

Visibilitats del mecanisme de comprovació d'identificador. Per tal de disposar d'un mecanisme de *comprovació d'identificador*, necessàriament el sistema (o més concretament, el controlador) ha de tenir visibilitat (directa o indirecta) de l'àmbit d'identificació.¹⁷

Àmbits accessibles. El controlador ha de tenir visibilitat (directa o indirecta) de l'àmbit d'identificació de tots i cadascun dels identificadors que apareixen com a arguments dels ES que captura.

Principi



Àmbit accessibles i identificació referencial. El principi dels *Àmbits accessibles*, que és conseqüència immediata del principi de la *Identificació robusta*, assegura les visibilitats necessàries pel principi de l'*Encaixada*. Per tant si es compleixen les condicions dle principi dels *Àmbits accessibles* tenim assegurades les exigències del principi de la *Identificació referencial*.

Àmbits accessibles i comprovació d'identificació. Les condicions del principi dels *Àmbits accessibles* assegura les condicions del principi de la *Identificació referencial*. Però si donat un identificador qualsevol sempre el podem encaixar, llavors sempre podem comprova l'existència de l'objecte referenciat: si la recuperació referencial té èxit llavors també l'ha de tenir la comprovació referencial;

¹⁷L'àmbit d'identificació l'hem definit com el repositori sobre el qual s'ha complir la unicitat. Per tant el que exigim és que el controlador tingui visibilitat sobre l'origen de l'explosió sobre la que es defineix aquest repositori (recordem que un repositori és l'abast d'una explosió).

altrament no. Per tant, les condicions del principi dels *Àmbits accessibles* asseguren que les exigències del principi de la *Identificació robusta* són realitzables.

(87)

El principi dels *Àmbits accessibles* subsumeix el principi de la *Identificació referencial*, i posa les condicions per poder complir amb el principi de la *Identificació robusta*

Principis i definicions del capítol

Principis

Consistència individualitzadora,
377

Estabilitat d'identificació, 380

Identificació atributiva, 380

Àmbits accessibles, 381

Definicions

Associació d'identificació, 369

Concepte feble, 368

Concepte fort, 369

Migració de realitzacions, 378

Realització compartida, 378

Repositori replicat, 379

Rèplica d'un repositori, 379

Àmbit d'identificació, 376

Capítol 20

Generació d'identificadors

1	Mecanismes de generació	391
1.1	Generació única	391
1.2	Generació per comprovació (GPC)	391
2	Introducció a la GPC	392
2.1	Recodatori sobre els repositoris i els àmbits d'identificació	392
2.2	La importància de l'àmbit d'identificació	394
3	GPC amb un sol repositori	399
3.1	Individualització tardana	399
3.2	Autoindividualització	403
3.3	Individualització genètica	404
3.4	Individualització induïda	406
3.5	GPC amb repositori únic: conclusions	409
4	GPC amb múltiples repositoris	409
4.1	Context de l'anàlisi	409
4.2	Repositori d'emmagatzematge diferent del de contrast	410
4.3	Replicació i fragmentació del repositori de contrast	411
5	Generació per comprovació: conclusions	413
6	Generació única	413
6.1	Plantejament de la generació única	413
6.2	Individualització tardana	415
6.3	Autoindividualització	415
6.4	Individualització induïda	416
6.5	Individualització genètica	416
7	Principi de la Individualització efectiva	416
	Principis	418
	Definicions	418

Contingut detallat del capítol 20

1 Mecanismes de generació	391
Dos mecanismes possibles	391
El camí que seguirem	391
1.1 Generació única	391
Valor independent del repositori	391
1.2 Generació per comprovació (GPC)	391
1.2.1 Responsabilitats de la GPC	391
Comprovació de la unicitat	391
Responsabilitats per a la generació per comprovació	391
On s'ha de fer la comprovació	392
Repetibilitat dels candidats	392
2 Introducció a la GPC	392
2.1 Recodatori sobre els repositoris i els àmbits d'identificació	392
2.1.1 Definicions bàsiques	392
Àmbit d'identificació	392
Repositori	392
Explosió i expansió	392
Expressió d'una explosió	392
Enllaços dirigits d'una explosió	392
Explosió o repositori	393
Expressió d'un àmbit de visibilitat	393
Nivells del repositori	393
Repositori d'un sol nivell	393
Amplitud d'un repositori	393
Representació gràfica d'un repositori	393
2.1.2 Tipus de repositori	393
Un repositori que en captura un altre	393
Repositoris germans	393
Repositori màxim	393
Repositori total	393
Repositori centralitzat	393
Recordatori. Repositori fragmentat	394
Fragments d'un repositori	394
Repositori centralitzat com a cas particular de fragmentació	394
2.1.3	394
Identificació global i local	394
2.2 La importància de l'àmbit d'identificació	394
Hipòtesi de treball	394
Identificació única	394
2.2.1 L'àmbit d'identificació és un repositori centralitzat	395
Una comprovació simple	395
Unicitat assegurada	395

	Compartició i migració	395
2.2.2	L'àmbit d'identificació és un fragment de repositori	395
	Identificació independent del repositori	395
	Migracions i comparticions	395
	Rèpliques	395
	Repositori de contrast i àmbit d'identificació	396
	Dependència del repositori de contrast	396
	Identificació local	396
2.2.3	Repositori de contrast centralitzat	396
2.2.4	Un fragment com a repositori de contrast	396
2.2.5	Experteses i experts en la generació per comprovació	396
	Expertesa de generació de candidat	396
	Expert en la comprovació de la identificació	397
	Experts en la generació per comprovació	397
	Assumpció simplificadora	397
	Validesa de l'assumpció	397
2.2.6	Repositori d'emmagatzematge	398
	Repositori d'emmagatzematge	398
	Repositori d'emmagatzematge i repositori de contrast	398
2.2.7	Àmbit d'identificació	399
	Localitat o globalitat de la identificació?	399
	Assumpció de treball	399
3	GPC amb un sol repositori	399
3.1	Individualització tardana	399
3.1.1	El mecanisme de la individualització tardana	399
	Generació en la inserció al repositori	399
	Aplicació de la <i>Identificació atributiva</i>	399
	Aplicació d' <i>Expert</i>	399
	Mecanisme d'individualització tardana	400
	Perill d'extraviar-se	400
3.1.2	Principi de la <i>Creació complerta</i>	400
	Individualització tardana i incompletesa	400
	Identificació atributiva	400
	Incompletesa de les realitzacions	400
	Creació complerta	400
3.1.3	Admissibilitat de la <i>Creació complerta</i>	401
	Exigència excessiva?	401
	Valor buit i inexistència de valor	401
	<i>Creació complerta</i> i valors buits	401
	Supressió dels valors buits	402
3.1.4	Conclusions sobre la individualització tardana	403
	Individualització tardana desaconsellable	403
3.2	Autoindividualització	403
3.2.1	El mecanisme de l'autoindividualització	403
	Generació en la creació	403
	Expert en la identificació	403

	Els principis de creació aplicats a l'identificador	403
3.2.2	Visibilitat del repositori de contrast	403
	Arguments de <code>create()</code>	403
	Alt acoblament; circularitat de dependències	404
3.2.3	Principi de <i>Desconeixement del grup</i>	404
	Desconeixement del grup	404
	<i>Desconeixement del grup</i> i repositori de contrast	404
3.2.4	Conclusions sobre l'autoindividualització	404
	Autoindividualització desaconsellable	404
3.3	Individualització genètica	404
3.3.1	El mecanisme d'individualització genètica	404
	El <i>creador</i> com a generador d'identificadors	404
	Mecanisme d'individualització genètica	404
	Visibilitat (quasi) garantida	405
	Mecanisme d'individualització genètica revisat	405
3.3.2	Delegació de la creació del valor d'identificació	405
	Delegació en l'objectiu	405
	Delegació en el repositori de contrast	405
	Assumpció de la responsabilitat	405
	Possible violació d' <i>Alta cohesió</i>	405
3.3.3	Conclusions sobre la individualització genètica	406
	Individualització genètica aconsellable	406
3.4	Individualització induïda	406
3.4.1	El mecanisme de la individualització induïda	406
	Transmissió al <i>creador</i> del valor d'identificació	406
	Mecanisme d'individualització induïda	406
	Acoblaments de la <i>individualització induïda</i>	406
	Necessitat de restriccions per a la consistència	406
3.4.2	Inconsistències en la identificació induïda	407
	Context de l'anàlisi	407
	Valors repetibles	407
	Valors no repetibles	407
	Excursió. (Múltiples individualitzadors)	407
	Conclusions sobre els problemes de consistència	407
3.4.3	Conclusions sobre la individualització induïda	408
	Individualització induïda desaconsellable	408
	Un mica de nomenclatura	408
	Perills del desacoblament extrem	408
	Nous arguments per a la <i>individualització genètica</i>	408
3.5	GPC amb repositori únic: conclusions	409
	No a la individualització tardana	409
	No a l'autoindividualització	409
	No a la individualització induïda	409
	Problemes de consistència amb la individualització induïda	409
	Sí a la individualització genètica	409
4	GPC amb múltiples repositoris	409

4.1	Context de l'anàlisi	409
	Múltiples repositoris	409
	Reordatori sobre els àmbits d'identificació	409
	Mantenim la hipòtesi de treball	409
	Experts en la generació per comprovació	410
4.2	Repositori d'emmagatzematge diferent del de contrast	410
4.2.1	Individualització tardana	410
	Violació de <i>Creació completa</i>	410
	Altres problemes amb múltiples repositoris	410
4.2.2	Autoindividualització	410
4.2.3	Individualització induïda	411
4.2.4	Individualització genètica	411
	Necessitat del repositori de contrast	411
	Visibilitats del creador orb	411
4.3	Replicació i fragmentació del repositori de contrast	411
4.3.1	Necessitat d'emmagatzematge múltiple	411
	Supòsit de l'anàlisi	411
	Utilitat d'una rèplica centralitzada	411
	Sincronització de repositoris	411
	Múltiple emmagatzematge	412
	Altres emmagatzematges	412
4.3.2	Responsable de l'emmagatzematge múltiple	412
	Acoblament i cohesió dels emmagatzematges múltiples	412
	El <i>peticionari</i> com a coordinador	412
	Excursió	412
5	Generació per comprovació: conclusions	413
	Mecanisme d'individualització	413
	Funcionament del mecanisme	413
	Visibilitats	413
6	Generació única	413
6.1	Plantejament de la generació única	413
6.1.1	Mecanismes per a la generació única	413
	Generació única	413
	Mecanismes de generació única	413
	Generació per seqüència infinita	413
	Generació aleatòria	414
	Generació pseudo-aleatòria	414
6.1.2	Estat intern de la generació	414
	Estat de la generació	414
	Experts en la generació única	414
	Extensió notacional	414
	Àmbit d'identificació	414
6.2	Individualització tardana	415
	Es mantenen els contres	415
	Visibilitats exigides	415
6.3	Autoindividualització	415

Desapareix un inconvenient	415
Arguments de <code>create()</code>	415
Acoblament i cohesió	415
6.4 Individualització induïda	416
Anàlisi anterior no aplicable	416
El llarg viatge dels identificadors	416
6.5 Individualització genètica	416
Espertesa perduda	416
Mantenim les conclusions	416
7 Principi de la <i>Individualització efectiva</i>	416
Individualització efectiva	416
Principis	418
Definicions	418

1 Mecanismes de generació

Com pot el sistema crear identificadors? Qui n'és el responsable?

(60) ?

Dos mecanismes possibles. Considerarem dos possibles mecanismes per a la generació del valor d'identificació:

- *Generació única*
- *Generació per comprovació*

El camí que seguirem. En aquest apartat presentem aquests dos mecanismes. En els apartats següents analitzem amb deteniment cadascun d'ells. Començarem per la *Generació per comprovació*, i l'analitzarem en diferents escenaris, cadascun més complex que l'anterior. A continuació analitzarem la *Generació única*. I finalment extraurem unes conclusions vàlides sigui quin sigui el mecanisme de generació d'identificador emprat.

1.1 Generació única

Valor independent del repositori. En cas de voler crear un valor d'identificació de manera independent al repositori ens caldria un mecanisme capaç de generar un valor diferent cada cop que se l'utilitza.

Exemple 87 (Generació única) *Un valor d'identificació possible pot ser el nombre de realitzacions creades fins el moment.*

Una altra possibilitat és la generació de combinacions de símbols de manera ordenada (per exemple, enumeració de Σ^ segons l'ordre lexicogràfic).¹*

Cal observar com, en ambdós exemples, internament cal guardar el darrer valor generat.

1.2 Generació per comprovació (GPC)

1.2.1 Responsabilitats de la GPC

Comprovació de la unicitat. Un mecanisme alternatiu a la *generació única* és la generació d'un valor (de manera aleatòria, per exemple) i després comprovar la seva unicitat en l'estat actual del repositori. És a dir, enlloc d'assegurar que el valor generat és nou, el que fem és generar un valor qualsevol, i tot seguit comprovar que en el repositori no hi ha cap realització identificada amb aquest valor d'identificació.

¹ Σ és un alfabet (conjunt de símbols) qualsevol. Σ^* és el conjunt de totes les combinacions possibles amb els símbols de Σ , d'un o més símbols. L'ordre lexicogràfic és l'ordre equivalent d'ordenació en un diccionari. Per exemple, "abc" va abans de "abd".

Responsabilitats per a la generació per comprovació. Així, la generació d'un valor d'identificació requereix resoldre dues responsabilitats diferents: per una banda cal crear el nou valor d'identificació (*Generació de candidat*); i per l'altra cal fer la comprovació de la unicitat d'aquest valor (*Comprovació de la identificació*).

On s'ha de fer la comprovació. En quin repositori cal fer la *comprovació d'identificació*? Doncs en aquell on no hi pugui haver dos objectes amb el mateix identificador. I això és l'àmbit d'identificació. →

19.4.1. Àmbit d'identificació, pàgina 375

(88)

Responsabilitats de la GPC.

- **Generació de candidat.** Crear un valor d'identificació.
- **Comprovació de la identificació.** Comprovar que aquest valor és inexistent en l'àmbit d'identificació. →

19.4.1. Àmbit d'identificació, pàgina 375

Repetibilitat dels candidats. La *generació de candidat* ha de poder generar valors repetits; és la *comprovació de la identificació* qui evita la generació de valors d'identificació repetits. Si la *generació de candidat* no pot generar valors repetits, la *comprovació d'identificació* esdevé supèrflua; de fet llavors estem en un cas de *generació única*, i no pas de *generació per comprovació*.

2 Introducció a la GPC

2.1 Recodatori sobre els repositoris i els àmbits d'identificació

2.1.1 Definicions bàsiques

Àmbit d'identificació. L'àmbit d'identificació d'un objecte $o : \circ$ és el repositori on el valor d'identificació de $o : \circ$ ha de ser únic. Si usem el principi de la *Identificació atributiva* llavors l'àmbit d'identificació d'un objecte $o : \circ$ és el repositori on $o . id$ és únic.

Repositori. Un *repositori* és l'abast d'una explosió. S'expressa indicant l'origen de l'explosió i l'expansió que l'explosó realitza. Sovint, però, l'expansió es dona per sobreentesa.

Explosió i expansió. Una *explosió* és la realització d'una expansió. Una *expansió* és una seqüència enllaçable de visibilitats generalitzades; la hipotètica inversa d'aquesta seqüència és una visibilitat indirecta monoavaluada.

Expressió d'una explosió. Una explosió es defineix indicant l'expansió de la qual és realització, i l'origen de l'explosió.

Enllaços dirigits d'una explosió. L'origen d'una explosió expressa el primer enllaç dirigit de l'explosió; tots els altres enllaços dirigits queden perfectament determinats per les visibilitat que configuren l'expansió

Explosió o repositori. A tots els efectes un repositori i una explosió són el mateix; el que els diferencia és què ens interessa realment de l'explosió: tota ella, el seu origen, o el seu abast.

Expressió d'un àmbit de visibilitat. Un *àmbit de visibilitat* s'expressa amb un objecte i una expansió. L'objecte és l'origen d'una explosió que realitza l'expansió donada; l'àmbit de visibilitat és l'abast d'aquesta explosió.

Nivells del repositori. El *nivell* d'un repositori és el nombre de visibilitats que conté l'expansió de la qual n'és realització.

Repositori d'un sol nivell. El repositoris d'un sol nivell es corresponen a una única visibilitat, generalment multiavaluada (i per tant expressable en termes de *multiobjecte*).²

Amplitud d'un repositori. Un repositori total és aquell qui conté totes les realitzacions que d'un determinat component hi ha en el sistema. Un repositori parcial només conté algunes de les realitzacions del component.

Representació gràfica d'un repositori. Un repositori és una graf dirigit acíclic amb una única arrel; l'alçada del graf és el nivell del repositori.

2.1.2 Tipus de repositori

Un repositori que en captura un altre. Donat un repositori x direm que *captura* el repositori y si i només si y és un subarbre de x . En cas de captura, l'expansió que realitza el repositori capturat és un infix de l'expansió que realitza el repositori captador.

Repositoris germans. Dos repositoris són *germans* si i només si hi ha un tercer repositori que els captura, i en aquest tercer repositori són subarbres del mateix nivell. Dos germans són realitzacions diferents d'una mateixa expansió.

Repositori màxim. Quan parlem de repositori ens referim al *repositori màxim*. Això és, al que correspon a l'expansió amb més visibilitats. Per tant, un repositori màxim mai no pot ser capturat.

Repositori total. Un *repositori total* és aquell qui conté totes les realitzacions, actuals i futures, que el sistema té del component en qüestió.

²La definició d'explosió, però, no exigeix que la darrera visibilitat sigui multiavaluada, i per tant no és correcte assimilar els repositoris d'un nivell als multiobjectes, tot i que la majoria sí que ho són.

Repositori centralitzat. Un *repositori centralitzat* és total i d'un sol nivell. Tot repositori centralitzat realitza una expansió que és un prefix (no necessàriament propi) de l'expansió que realitza un repositori màxim.

Recordatori. Repositori fragmentat. Un *repositori fragmentat*³ és aquell en què totes les visibilitats de l'expansió, a excepció de la primera, són obligades; i totes les hipotètiques inverses de les visibilitats de l'expansió són obligades.

Fragments d'un repositori. En un *repositori fragmentat* cada objecte no fulla de l'explosió és l'origen d'un *fragment del repositori*. Tots els fragments del mateix nivell diem que són germans. Els fragments germans són disjunts. Un repositori fragmentat és un arbre amb totes les branques amb la mateixa alçada.

Repositori centralitzat com a cas particular de fragmentació. Tot repositori centralitzat és un cas particular de repositori fragmentat: consta d'un sol fragment, i aquest consta d'un sol nivell.

2.1.3

Identificació global i local. Si l'àmbit d'identificació és un repositori total diem que la identificació és *global*; si és el fragment d'un repositori total diem que la identificació és *local*.

. Per tal que un repositori pugui ser un àmbit d'identificació cal que els àmbits d'identificació sobre un mateix component :

- Si són del mateix nivell han de ser disjunts.
No és necessari, però ajuda
- Els àmbits d'un nivell poden ser capturats per altres àmbits

2.2 La importància de l'àmbit d'identificació

Hipòtesi de treball. De moment considerem que l'àmbit d'identificació es materialitza en un sol repositori, que no té rèpliques.

H (61)

- *Identificació única.* Tot objecte només pertany a un àmbit d'identificació
- *Identificació no replicada.* L'àmbit d'identificació correspon a un únic repositori; és a dir, no admetem rèpliques \rightarrow de l'àmbit d'identificació

Si l'àmbit d'identificació és, per exemple, un repositori centralitzat de A , llavors no admetem que hi pugui haver cap repositori fragmentat de A .

³Formalment, fragmentat per nivells sencers

Identificació única. L'exigència de la identificació única diu que cada objecte té un sol valor d'identificació. Aquest supòsit és del tot coherent amb els principis de l'*Estabilitat identificadora* i de la *Identificació atributiva*: tot objecte té un únic identificador, que n'és una pròprietat intrínseca no modificable.

2.2.1 L'àmbit d'identificació és un repositori centralitzat

Una comprovació simple. Suposem que tenim un *repositori centralitzat* dels objectes que volem identificar. Res impedeix que considerem aquest repositori com l'àmbit d'identificació. I llavors, per fer la *comprovació d'identificació* d'un nou identificador n'hi ha prou en fer un `find()` sobre aquest repositori.

GPC amb repositori centralitzat com a àmbit d'identificació.

Un mecanisme de generació d'identificadors, per comprovació, sobre un àmbit d'identificació que és un *repositori centralitzat*, proporciona una identificació d'*àmbit global*, i *admet* tant les *migracions* com les *comparticions* de realitzacions.

(89)

Unicitat assegurada. Tota realització $c : C$ del sistema està en el repositori de contrast R . La generació per comprovació sobre R assegura que no es generi cap identificador repetit a R . Per tant no és possible que el sistema tingui dues realitzacions diferents $c_1 : C$ i $c_2 : C$ amb el mateix identificador.

Compartició i migració. L'àmbit d'identificació global assegura que dues realitzacions no poden tenir el mateix identificador, i per tant no hi ha cap perill en canviar un objecte de repositori, o compartir-lo en un altre repositori.

2.2.2 L'àmbit d'identificació és un fragment de repositori

Identificació independent del repositori. Donat un component C l'objectiu és definir un mecanisme de *generació d'identificació* cada objecte $c : C$.

GPC amb un fragment de repositori com a àmbit d'identificació.

Un mecanisme de generació d'identificadors, per comprovació, sobre un àmbit d'identificació que és un *fragment d'un repositori*, proporciona una identificació d'*àmbit local* a aquest fragment, i en general *no admet* ni les *migracions* ni les *comparticions* de realitzacions.

(90)

Migracions i comparticions. La migració o la compartició de realitzacions en general no manté les propietats d'unicitat de l'identificador: que un valor sigui inexistent en un repositori (i per tant el podem usar per identificar el nouvingut) no significa que no pugui figurar en un altre repositori (i que per tant no ens serveixi com a valor per identificar una nova realització).

Rèpliques. En el cas de les rèpliques, com que contenen exactament les mateixes realitzacions, si un identificador és únic en una rèplica, també ho ha de ser en l'altra.

Definició. Repositori de contrast

En els mecanismes de generació d'identificadors per comprovació anomenem repositori de contrast aquell multiobjecte sobre el qual el mecanisme assegura unicitat del valor creat.⁴

Repositori de contrast i àmbit d'identificació. El *repositori de contrast* és un *multiobjecte*; és a dir, un repositori d'un sol nivell. En canvi l'*àmbit d'identificació* pot ser qualsevol repositori.

Dependència del repositori de contrast. El mecanisme generador dels identificadors per comprovació depèn del repositori sobre el que es fa la comprovació. El mecanisme de generació única n'és del tot independent.

Identificació local. La generació per comprovació genera identificadors vàlids en àmbit local del repositori de contrast.

2.2.3 Repositori de contrast centralitzat

2.2.4 Un fragment com a repositori de contrast

2.2.5 Experteses i experts en la generació per comprovació

Expertesa de generació de candidat. Els elements que intervenen en la generació d'un candidat pel valor d'identificació depenen de cada problema, tot i que en general es poden considerar dues alternatives extremes:

- Els **identificadors significatius o analítics** són aquells valors d'identificació que tenen un significat analític per a l'usuari final; en aquest cas la informació d'expertesa són els arguments de l'ES pertinent.
- Els **identificadors interns o sintètics** són aquells valors d'identificació que no tenen cap mena de significat per a l'usuari final, és a dir, es tracta de valors generats internament pel sistema; en aquests casos la informació d'expertesa depèn de com s'hagi dissenyat aquest mecanisme.

Exemple 88 (Identificador analític i sintètic)

- Identificador analític. *MNY-PG-Corredor*. Identificador per una *c*: *Caminada classificada com una passejada* (*PG=passeig*); que transcorre per la zona del Montseny

⁴El repositori de contrast i l'àmbit d'identificació són el mateix concepte, però vistos des d'un punt de vista diferent. L'àmbit d'identificació focalitza en la propietat d'unicitat; el repositori de contrast, que és un concepte més mecanicista, focalitza en el mecanisme per obtenir la unicitat.

(MNY=Montseny), i més concretament per la serra del Corredor.

- Identificador sintètic. `34er5TgGH2`. Valor generat internament pel sistema i que es podria usar per a identificar una `c`: Caminada qualsevol (per exemple, el passeig per la serra del Corredor).

Expert en la comprovació de la identificació. El propi repositori de contrast és clarament un expert per a la comprovació de la identificació.⁵

Experts en la generació per comprovació. Per ser expert en la generació per comprovació cal ser expert en la generació del valor d'identificació, i expert en la comprovació de la identificació. Així, tot component que tingui visibilitat del repositori de contrast també és un possible expert en la generació per comprovació de l'identificador: n'hi ha prou en assegurar que és un expert en la creació o generació del candidat a valor d'identificació, i llavors delegar en el repositori la comprovació d'unicitat.

Assumpció simplificadora. Assumim que tot expert en la comprovació també és un expert en la creació d'un candidat a valor d'identificació. Aquesta assumpció permet centrar tota l'anàlisi en termes de la visibilitat sobre el repositori de contrast: segons ella, el coneixement del repositori de contrast és la condició necessària i suficient per tal que un component esdevingui expert de la generació per comprovació.

Expert en la comprovació d'identificació \iff Expert en la generació de candidat

(62) ? !

Validesa de l'assumpció. Com hem vist l'expertesa per a la generació per comprovació requereix l'expertesa en la generació del valor d'identificació, i l'expertesa en la comprovació de la unicitat d'aquest valor. L'assumpció simplificadora considerada considera les dues experteses com a una de sola. En cas de ser falsa l'assumpció llavors l'expert considerat en tota l'anàlisi que segueix és només un expert parcial per a la generació per comprovació. Llavors:

- Si a aquest expert parcial li podem delegar l'expertesa en la generació d'un candidat d'identificació, tota l'anàlisi continua sent vàlida.
- Si la delegació no és possible caldrà cercar algú que coordini les dues experteses parcials.
- Sigui com sigui, els resultats obtinguts en l'anàlisi que segueix continuaran essent vàlids, i com a molt caldrà afegir les noves restriccions.

⁵Un repositori és l'abast d'una explosió, i s'expressa amb l'objecte origen de l'explosió. Per tant el que estem dient aquí és que l'expert GPC és l'objecte origen de l'explosió que defineix el repositori de contrast.

2.2.6 Repositori d'emmagatzematge

Repositori d'emmagatzematge. L'ús del principi *Creador orb* assegura que el *creador* de l'objectiu de la identificació és el responsable d'emmagatzemar-lo; i per tant té visibilitat del repositori d'emmagatzematge.

Repositori d'emmagatzematge i repositori de contrast. Tota inserció que es faci en el repositori d'emmagatzematge també s'ha de fer en el de contrast per mantenir la consistència de la generació per comprovació. Per tant el repositori d'emmagatzematge ha de ser un subconjunt del repositori de contrast.

Repositori d'emmagatzematge.

El repositori d'emmagatzematge ha de ser un subconjunt del repositori de contrast

(91)

2.2.7 Àmbit d'identificació

Localitat o globalitat de la identificació? En l'anàlisi de la generació per comprovació ens cal tenir en compte el repositori de contrast. Ara bé, per assegurar que la generació per comprovació es comporta correctament és indiferent si l'àmbit d'identificació que defineix el repositori de contrast és local o global: el que cal és assegurar que mai no es generarà un valor existent en el repositori de contrast.

Assumpció de treball. Per seguir en l'anàlisi considerarem que el repositori de contrast és centralitzat i sense rèpliques. Un cop acabada l'anàlisi analitzarem[→] la influència que pot tenir un repositori de contrast fragmentat (i per tant la identificació d'àmbit local) o un repositori replicat.

20.4.3.Replicació i fragmentació del repositori de contrast, pàgina 411

En el que segeueix considerem que el repositori de contrast és centralitzat i sense rèpliques.

(63) ? !

3 Generació per comprovació amb un sol repositori

Considerem el cas en què l'objectiu d'identificació només intervé en un repositori. És a dir, el repositori d'emmagatzematge i el repositori de contrast coincideixen.

3.1 Individualització tardana

3.1.1 El mecanisme de la individualització tardana

Generació en la inserció al repositori. El moment culminant de l'expertesa en la generació del valor d'identificació és en el moment de la inserció de l'objectiu en el repositori: és el moment en el que sabem exactament de quin conjunt cal individualitzar-nos. De fet aquest és el moment més tardà possible per generar el valor d'identificació; és el moment més tardà per individualitzar una realització concreta del seu col·lectiu.

Aplicació de la Identificació atributiva. La individualització tardana sembla que viola el principi de la *Identificació atributiva*[→] ja que lliga el valor de l'identificador amb el repositori on s'insereix l'objecte. Però sota la hipòtesi que estem considerant d'un sol repositori no hi ha tal violació.

19.5.Identificació atributiva, pàgina 379

Aplicació d'Expert. El repositori de contrast,⁶ com a *expert en la comprovació de la identificació*, rep la responsabilitat de generar el valor d'identificació.⁷

Individualització en l'add()

Mecanisme d'individualització tardana. La unicitat del repositori permet usar el mecanisme d'inserció per exercir la responsabilitat de generar el valor d'identificació.⁸ És a dir, el repositori de contrast (que és l'únic repositori en el que participa l'objecte considerat) ha de generar l'identificador d'un objecte en el moment que aquest s'introdueix en el repositori.⁹ Per tant, el mètode `add()`, que rep la realització a inserir al repositori, ha de generar l'identificador de l'objecte rebut i ha d'inserir aquest objecte.

Perill d'extraviar-se. El primer problema amb aquesta solució és que caldrà afegir un mecanisme per obtenir l'identificador de la realització acabada d'inserir. Altrament mai més no la podrem recuperar mitjançant un `find()` sobre el repositori de contrast utilitzat.

3.1.2 Principi de la Creació complerta

Individualització tardana i incompletesa. L'altra problema de la *individualització tardana* és més subtil: demana l'existència d'objectes temporalment incomplets. Tot seguit ho analitzem.

Identificació atributiva. El principi de la *Identificació atributiva* diu que la identificació es modelitza amb un atribut del component pertinent. És a dir, que donat un component `C` i un atribut `ID` de `C`, el valor que identifica la realització `c : C` és el valor que `c : C` dona a l'atribut `ID`.

Incompletesa de les realitzacions. Si el valor de l'identificador es genera en inserir la realització en el repositori de contrast, vol dir que des de la creació de la realització fins el moment de la inserció l'atribut d'identificació no tenia valor. En d'altres paraules, la realització estava incompleta.

Principi



Creació complerta. La creació d'un objecte l'ha de crear per complert; no es poden deixar atributs sense valor.

⁶Concretament, l'origen de l'explosió que defineix aquest repositori.

⁷Recordem l'assumpció simplificadora \rightarrow) segons la qual l'expertesa en la comprovació de la unicitat és condició necessària i suficient per l'expertesa en la generació per comprovació.

⁸Donar la responsabilitat de la indivualització a l'operació d'inserir en el repositori només té sentit si la inserció es fa sobre el repositori de contrast. La unicitat del repositori assegura aquesta condició.

⁹Més concretament, és l'origen de l'explosió que defineix el repositori qui ha de generar l'identificador.

3.1.3 Admissibilitat de la Creació complerta

Exigència excessiva? El principi de la *Creació complerta* pot semblar excessivament restrictiu. Vegem-ne un exemple:

Exemple 89 (Creació complerta i optativitats) *Suposem una associació entre A i B amb optativitat a l'extrem de B.*

Suposem que hem decidit modelitzar l'associació amb una visibilitat d'atribut $A \rightarrow B$. Suposem que implemtem aquesta visibilitat amb un atribut de A de nom $quinB$.

Això significa que totes aquelles realitzacions del concepte A que no s'hagin d'enllaçar a cap realització de B les modelitzarem amb objectes $a : A$ tals que el valor del seu atribut $quinB$ haurà d'estar indefinit (no ha de tenir valor).

Sembla doncs que es viola el principi de la Creació complerta: hi ha realitzacions de $a : A$ que no es poden construir donant valor a tots els seus atributs.

Valor buit i inexistència de valor. En tot moment cal saber distingir quan el valor d'un atribut és buit (o nul), de quan no hi ha valor. Els atributs modelen propietats; un valor buit indica que tenim el coneixement que cap dels valors possibles compleix la propietat. La inexistència d'un valor es dona quan en un atribut encara no se li ha donat cap valor (ni tant sols el valor buit) .

Exemple 90 (Valor buit i inexistència de valor) *Tothom viu en algun lloc, per tant tothom té adreça. Sota aquest supòsit un atribut que modela l'adreça mai no podrà tenir valor buit. Una altra cosa és que en el moment d'introduir les dades d'una persona al sistema en desconeguem l'adreça. Llavors estem parlant d'una adreça desconeguda, no pas d'una adreça buida.*

Hi ha persones que no tenen telèfon. Per tant és del tot lícit donar el valor buit a l'atribut que modela el número de telèfon. Així el número de telèfon podrà ser conegut (i indicarem quin és), desconegut (tot i que sabem que en té) o buit (amb el coneixement actual entenem que no en té).

Creació complerta i valors buits. La *Creació complerta* exigeix donar valor a tots els atributs; però un valor possible és el buit. El que intenta evitar la *Creació complerta* és l'existència en el sistema d'objectes mig construïts; en cas d'admetre-ho caldria afegir tota una sèrie de controls (via PRE dels contractes; via comprovacions) per assegurar que les operacions sobre els objectes només es realitzen quan aquests estan en la fase de creació necessària per aquesta a operació.

Exemple 91 (Problemes amb els valors buits) *Si permetem crear una $p : Persona$ sense adreça, llavors per usar el mètode `ciutatOnViu`*

sobre $p:Persona$ caldrà assegurar que en el moment d'enviar el missatge $p:Persona$ ja té una adreça introduïda.

Supressió dels valors buits. Sovint una manera d'evitar haver de treballar amb valors buits és a través d'una modificació del MC.¹⁰ Un exemple pot ser la reconversió de propietats condicionals en propietats obligades.

Exemple 92 (Modelització de les optativitats) El concepte *Persona* s'associa a *Telèfon* (via l'associació *contactable*). Aquesta associació és optativa per l'extrem diferent de *Persona*. Suposem que l'associació *contactable* és $1 - 1$ (una persona pot tenir com a molt un telèfon).¹¹

Si afegim un valor "No en té" com a possible realització del concepte *Telèfon* llavors l'associació *contactable* passa a ser obligada en l'extrem diferent de *Persona*. Però això exigeix que la multiplicitat a l'extrem de *Persona* sigui N : com a mínim el valor No en té pot estar assignat, via *contactable*, a diferents persones. El resultat és que ha canviat la multiplicitat de l'associació; i que si no afegim restriccions semàntiques addicionals permet que un mateix telèfon sigui compartit per diferents persones.

Per solucionar-ho introduïm el concepte *Contactació* que substituirà el concepte *Telèfon* en el MC original; i exigim, tal com feiem amb *contactable*, que sigui obligada per l'extrem diferent de *Persona*. El concepte *Contactació* el podem veure com una llista de telèfons de mida com a molt 1. D'aquesta manera una persona sempre té una *contactació*, que pot ser una llista buida o unitària.

L'associació *contactable* continua sent $1 - 1$. Cal observar que hi poden haver moltes realitzacions diferents de *Contactació* que siguin una "llista buida" de telèfons: cada llista, buida o no, és una nova realització.

La modificació del model no només ha suprimit la necessitat de treballar amb valors buits, sinó que permet la generalització a múltiples telèfons per persona de manera totalment trivial.

Un altre avantatge de la proposta és que, si cal, podem distingir entre valor buit i valor inexistent. Si sabem que una persona no té telèfon l'enllacem a una *contactació* que és una llista buida de telèfons. Si d'aquesta persona sabem algunes dades, però ens hem oblidat de preguntar-li el telèfon, llavors simplement no l'enllacem a cap *contactació*.¹²

¹⁰En el moment de dissenyar la nostra visió del problema pot canviar. D'aquí que el MComp final potser s'allunyaria molt del que correspondria per l'aplicació del principi de l'*Espill*. Si volem ser fidels al principi de l'*Espill* cal modificar MC a mesura que la construcció de MComp ho vagi demanant. Sovint però s'aplica la modelització especular en les primeres fases del disseny, i es va relaxant a mesura que es va avançant, de tal manera que el MComp resultant s'assembla ben poc al MC.

¹¹Com sabem sovint les associacions $1 - 1$ són símptoma d'una mala modelització. Aquí podem justificar la seva presència gràcies a què amb ella evitem els atributs optatius.

¹²El problema és que ara hem reintroduït optativitats. Però potser paga la pena davant la nova expressivitat del model.

3.1.4 Conclusions sobre la individualització tardana

Individualització tardana desaconsellable. En conseqüència, sembla que el principi de la *Creació complerta* és raonable. Però aquest principi prohibeix, sota el supòsit de la *Identificació atributiva*, que el repositori sigui el responsable de la creació de l'identificador en el moment d'inserir l'objecte. Per tant desaconsella la *individualització tardana*.

3.2 Autoindividualització

3.2.1 El mecanisme de l'autoindividualització

Generació en la creació. El principi de la *Creació complerta* porta a pensar en la generació del valor d'identificació en el mateix moment que es crea l'objectiu de la identificació. És a dir, és el `create()` qui té la responsabilitat de crear un objecte juntament amb el seu identificador.

Expert en la identificació. Recordem que en la *generació per comprovació* l'expert en la creació dels valors d'identificació és el repositori de contrast.¹³

Els principis de creació aplicats a l'identificador. El principi de la *Identificació atributiva* diu que el valor de l'identificació s'emmagatzema dins del propi objectiu de la identificació. Llavors, sota el supòsit que l'objectiu de la identificació té visibilitat del repositori de contrast, podem aplicar el principi *Creador orb*: l'objectiu de la identificació és expert per delegació de la identificació (per la hipòtesi sobre la visibilitat del repositori);¹⁴ i n'és emmagatzemador (per la *Identificació atributiva*). L'aplicació del principi *Creador orb* ens proposa que sigui el mateix objectiu de la identificació qui generi el seu identificador; demana l'autoindividualització.

3.2.2 Visibilitat del repositori de contrast

Com assegurem que l'objectiu de la identificació tingui visibilitat sobre el repositori de contrast?

(64) ?

Arguments de `create()`. L'única possibilitat per assegurar la visibilitat del repositori de contrast és demanar que el creador de l'objectiu de la identificació el passi com a argument del mètode `create()`.¹⁵ Això és així perquè la

¹³Recordem l'assumpció simplificadora → segons la qual l'expertesa en la comprovació de la unicitat és condició necessària i suficient per l'expertesa en la generació per comprovació.

¹⁴La hipòtesi de visibilitat assegura que l'objectiu de la identificació és expert. La justificació que aquesta expertesa és delegada es veurà tot seguit.

¹⁵En d'altres paraules, la informació d'expertesa per a la generació per comprovació d'un valor d'identificació és delegada des del creador a l'objectiu. L'objectiu per tant és un expert delegat.

visibilitat global està prohibida; per tant, les úniques visibilitats que té l'objecte que s'està creant són les visibilitats de paràmetre del mètode `create()`.¹⁶

Alt acoblament; circularitat de dependències. El resultat és que acoblem les realitzacions al repositori. Per altra banda, el repositori clarament té visibilitat d'atribut de les realitzacions que emmagatzema; per tant en total s'obté una circularitat de dependència, que és preferible evitar.

3.2.3 Principi de *Desconeixement del grup*

Principi



Desconeixement del grup.

Les realitzacions desconeixen a quin grup pertanyen; és a dir, desconeixen a quin repositori s'emmagatzemen.

Desconeixement del grup i repositori de contrast. En la generació per comprovació cal conèixer el grup en el moment de la creació d'una realització. En concret cal conèixer el repositori de contrast sobre el que es vol definir l'àmbit d'identificació. El que diu el principi del *Desconeixement del grup* és que aquest coneixement es limita a aquí; un cop creada, la realització ja es pot oblidar del grup de pertinença.¹⁷

3.2.4 Conclusions sobre l'autoindividualització

Autoindividualització desaconsellable. El principi del *Desconeixement del grup* evita donar a l'objectiu la responsabilitat de generar els identificadors; per tant desaconsella l'autoindividualització.

3.3 Individualització genètica

3.3.1 El mecanisme d'individualització genètica

El creador com a generador d'identificadors. La *Creació completa* exigeix que com a resultat de la creació de l'objecte aquest tingui un valor d'identificació assignat. El principi del *Desconeixement del grup* desaconsella l'*autoindividualització* (és a dir, l'objectiu de la identificació com a responsable de la generació del valor d'identificació). Ens queda per analitzar que sigui el propi creador qui generi el valor d'identificació. D'aquesta manera en la *gènesi* de la nova realització obtenim aquesta i el seu valor d'identificació.

¹⁶En els llenguatges OOP els components s'implementaran amb classes. Alguns llenguatges, com el C++ o el Java, permeten tenir propietats de classe, a part de les tradicionals propietats d'objecte. Llavors és possible mantenir amb *visibilitat d'atribut* de la classe el repositori de contrast. Com a conseqüència no cal passar per paràmetre el repositori al mètode `create()` (és a dir, el constructor de l'objecte).

¹⁷Una altra cosa és que sota la hipòtesi de migracions o comparticions el disseny s'hagi d'assegurar que l'àmbit d'identificació es manté vàlid. És a dir, podrem migrar a un fragment diferent del repositori de contrast, per exemple, però no a un repositori genèric qualsevol.

Mecanisme d'individualització genètica. Per assignar la responsabilitat de generar els valors d'identificació al *creador* de l'objectiu a identificar, cal que el mètode que fa la petició de creació de l'objectiu tingui visibilitat del repositori de contrast. I això només és possible si el mètode rep el repositori de contrast per paràmetre, o bé el creador en té visibilitat d'atribut.

Visibilitat (quasi) garantida. L'ús del principi *Creador orb* assegura que el *creador* de l'objectiu és el responsable d'emmagatzemar la realització creada; i per tant té visibilitat del repositori d'emmagatzematge. Però, per hipòtesi, tenim un sol repositori on participa la nova realització; repositori que actua com a repositori d'emmagatzematge i com a repositori de contrast. Així l'ús del principi *Creador orb* assegura que el *creador orb* té visibilitat d'atribut del repositori de contrast, i per tant també pot ser el *creador del valor d'identificació*.

Mecanisme d'individualització genètica revisat. Sota la hipòtesi d'usar un *creador orb*, → aquest respondrà la petició de creació d'una nova realització creant primer l'identificador corresponent; a continuació emetrà el missatge `create()` que, entre d'altres coses, rebrà com a argument l'identificador que cal associar a la nova realització.

Principi pàgina 163. (10.3. Principi *Creador orb*)

3.3.2 Delegació de la creació del valor d'identificació

El *creador* (de l'objectiu i del valor d'identificació) com crea el valor d'identificació? Se n'encarrega directament o ho delega a algú?

(65) ?

Delegació en l'objectiu. La delegació de la creació del valor d'identificació en l'objectiu és l'autodidentificació: viola el principi del *Desconeixement del grup* (i per tant acobla l'objectiu al repositori de contrast).

Delegació en el repositori de contrast. Estem analitzant a qui pot delegar el creador la generació de l'identificador. Però com que e's un *creador orb*, també és el responsable de l'emmagatzematge de l'objectiu. A més treballem sota la hipòtesi que el repositori de contrast i el d'emmagatzematge és el mateix. Per tant el *creador orb* és l'origen de l'únic repositori que tenim, el d'emmagatzematge que també és de contrast. En conseqüència, que el *creador orb* delegui en el repositori de contrast no és cap delegació.

Assumpció de la responsabilitat. Si el *creador orb* (que és el repositori de contrast) assumeix la responsabilitat de generar l'identificador, és a dir, si no delega la responsabilitat, li disminueix la *Cohesió*.

Fins a quin punt la no delegació de la responsabilitat de creació del valor d'identificació disminueix la cohesió del creador?

(66) ?

Possible violació d'Alta cohesió. Els principis de la *Creació complerta* i de la *Identificació atributiva* diuen que la nova realització ha de tenir valor d'identificació. Per tant sembla lògic que el creador de l'objectiu sigui el responsable de crear aquest valor d'identificació. És a dir, podem considerar la creació del valor d'identificació com a part de la responsabilitat de crear l'objectiu; amb aquesta visió de les coses, la cohesió del creador no disminueix: mantenim les responsabilitats de creació i emmagatzematge; el que passa és que augmenten les sub-responsabilitats del que entenem per creació.

3.3.3 Conclusions sobre la individualització genètica

Individualització genètica aconsellable. La decisió d'assignar al *creador* de l'objectiu la responsabilitat de crear-ne el valor d'identificació sembla força consistent amb tots els principis de disseny. Les restriccions són que la responsabilitat s'ha d'exercir en el mètode de creació de l'objectiu, i que la tasca no pot ser delegada a l'objectiu.

3.4 Individualització induïda

3.4.1 El mecanisme de la individualització induïda

Transmissió al creador del valor d'identificació. El *creador* és expert en la creació de l'objectiu. Sovint aquesta expertesa és delegada:¹⁸ tota o part de la *informació de l'expertesa* es transmet al *creador* a través dels arguments del missatge de petició de la creació de la nova realització. Res impedeix transmetre també el valor d'identificació. D'aquesta manera el *creador* es veu forçat, induït, a crear una nova realització amb un determinat valor d'identificació.¹⁹

Mecanisme d'individualització induïda. El *creador* fa de pur transmissor del valor d'identificació: el rep en la petició de creació, i el transmet a `create()`. Però això significa que el *peticionari* ha de demanar a un *individualitzador* la creació d'un valor d'identificació; per tot seguit passar aquest valor al *creador*.

Acoblaments de la individualització induïda. Per tal d'exercir la seva responsabilitat, l'*individualitzador* ha de tenir visibilitat sobre el repositori de contrast; però sota les condicions del principi *Creador orb*[→] el creador també n'ha de tenir visibilitat.²⁰ La individualització induïda, per tant, augmenta l'acoblament cap el repositori de contrast. Per altra banda, el valor d'individualització queda acoblat a tots els components de les cadenes de delegació que van del *peticionari* a l'*individualitzador*, i del *peticionari* al *creador*.

10.3.Principi *Creador orb*, pàgina 161

10.2.2.Delegació de la creació, pàgina 160

¹⁸El principi *Creador delegat*[→] permet aquesta delegació de l'expertesa; i per conseqüent també els principis *Creador orb* i *Creador canònic*.

¹⁹El valor d'identificació, així, el podem veure com a part de la *informació d'expertesa*.

²⁰Aquesta afirmació només té sentit sota la hipòtesi de treball de repositori únic.

Necessitat de restriccions per a la consistència. En la individualització induïda assignem les responsabilitats de creació i individualització a components diferents i potencialment independents. Per tal d'assegurar el correcte funcionament del sistema caldria establir mecanismes per assegurar la correcta sincronització i consistència d'ambdues responsabilitats. Tot seguit justifiquem aquesta necessitat de consistència.

3.4.2 Inconsistències en la identificació induïda

Context de l'anàlisi. Suposem un context inicial en el que un *peticionari* p demana a l'*individualitzador* i la creació d'un valor d'identificació. Suposem que el valor obtingut és id . Suposem que el *peticionari* p "s'oblida" de demanar a cap *creador* la creació d'un *objectiu* identificat amb el valor id .²¹

Què passa si un altre *peticionari* q (o fins i tot el mateix p) demana un nou valor d'identificació a l'*individualitzador* i ?

(67) ?

Valors repetibles. Suoposem que es pugui obtenir el mateix valor id (el repositori de contrast ho permet: hem suposat que el primer cop que s'ha generat aquest valor d'identificació no hem creat cap objecte; i per tant no hem pogut afegir un objecte amb aquest valor d'identificació en el repositori de contrast). Així, el *peticionari* q pot crear (i emmagatzemar) un *objectiu* identificat pel valor id . Però llavors si en un altre moment posterior el *peticionari* p usa el valor id que li ha proporcionat l'*individualitzador* i per crear una nova realització, estarem demanant la creació d'un objecte amb un identificador que correspon a un altre objecte ja creat!!!

Valors no repetibles. Si suposem que no es pot obtenir el mateix valor id llavors l'*individualitzador* es converteix en un mecanisme d'identificació de *generació única*: cada petició d'un valor d'identificació proporciona un valor nou (per hipòtesi de la no repetibilitat), independentment de si després aquest valor d'identificació s'usa per crear i emmagatzemar un objecte o no.

Excursió. (Múltiples individualitzadors) Si admetem la possibilitat d'usar més d'un *individualitzador* per crear els identificadors dels objectes que aniran en un mateix repositori, caldrà exigir que el conjunt dels *individualitzadors* usats es comporti com un únic identificador. És a dir, que si un *individualitzador* i_1 genera un valor id s'ha de complir que cap altre *individualitzador* i_2 pugui generar el mateix valor id en un moment posterior. Cas de no ser així ens trobaríem amb els mateixos problemes que els exposats en l'anàlisi dels *valors repetibles*.

²¹L'anàlisi del cas en què p crea un objecte amb l'identificador id però s'"oblida" d'emmagatzemar-lo és similar i porta a les mateixes conclusions. Ho deixem com exercici pel lector.

Conclusions sobre els problemes de consistència. Si volem que la generació dels valors d'identificació sigui dependent del *repositori de contrast* (és a dir, si volem un mecanisme de *generació per comprovació*) l'*individualitzador* podrà repetir els valors d'identificació que no es corresponguin a cap objecte dels presents en el *repositori de contrast*. Però llavors els *peticionaris* hauran de fer comprovacions extres abans de crear i/o emmagatzemar noves realitzacions.

3.4.3 Conclusions sobre la individualització induïda

Individualització induïda desaconsellable. Convertir el *creador* en mer transmissor del valor d'identificació genera acoblaments amb el repositori de contrast i amb el propi valor d'identificació; i alhora deixa un forat obert a la inconsistència (si no afegim cap mecanisme de control es poden generar valors d'identificació que poden col·lisionar amb d'altres dels valors generats). Tot plegat porta a considerar la *individualització* induïda com a desaconsellable.

Un mica de nomenclatura. L'acoblament l'hem presentat en termes de components. Si abusem una mica de la terminologia, direm que dues responsabilitats estan acoblades quan les hem assignades al mateix component; dues responsabilitats direm que estan semànticament acoblades si per la descripció d'una cal fer referència a l'altra.

Perills del desacoblament extrem. En la *individualització induïda* donem la responsabilitat de la individualització i de la creació a components independents. En d'altres paraules, desacoblem ambdues responsabilitats. Ara bé, semànticament es tracta de responsabilitats estretament acoblades: la *generació per comprovació* assumeix²² que tots els objectes del col·lectiu del qual cal individualitzar un nou objecte estan presents en el *repositori de contrast*; i que tot valor d'identificació generat és realment usat per crear i emmagatzemar un nou objecte. En desacoblar la individualització de la creació no tenim manera (simple) d'assegurar la PRE exigida per la individualització.

Nous arguments per a la individualització genètica. Si per contra aquestes dues responsabilitats, que semànticament estan estretament acoblades, les assignem a un mateix component, tots aquests problemes desapareixen. El resultat és la *individualització genètica*: el *creador* rep les responsabilitats de generar el valor d'identificació per tot seguit crear el nou objecte; a més, si es tracta d'un *creador orb* també té la responsabilitat d'emmagatzemar el nou objecte.

(92)

La *individualització genètica* i el principi del *Creador orb* asseguren el correcte funcionament de la *generació per comprovació* de valors d'identificació (sota la hipòtesi de repositori únic).

²²És a dir, té com a PRE.

3.5 GPC amb repositori únic: conclusions

No a la individualització tardana. Els principis de la *Identificació atributiva* i de la *Creació complerta*^{→,→} prohibeixen la *individualització tardana*, és a dir, que el responsable de crear l'identificador sigui el repositori en el moment d'inserir.

19.5. Identificació atributiva, pàgina 379

20.3.1.2. Principi de la *Creació complerta*, pàgina 400

No a l'autoindividualització. El principi del *Desconeixement del grup*[→] prohibeix que el responsable en sigui el propi *objectiu*; per tant l'*autoindividualització* queda descartada.

20.3.2.3. Principi de *Desconeixement del grup*, pàgina 404

No a la individualització induïda. El principi *Baix Acoblament*[→] desaconsella que el valor d'identificació arribi al *creador* via arguments; per tant la *individualització induïda* no apareix com un mecanisme aconsellable.

8.3. Acoblament, pàgina 128

Problemes de consistència amb la individualització induïda. Els problemes de possibles inconsistències derivades d'un desacoblament extrem són un argument molt fort per refusar la *individualització induïda*.

Sí a la individualització genètica. La *individualització genètica* apareix com el mecanisme més aconsellable. Donar la responsabilitat de generar el valor de l'identificador al *creador* de l'*objectiu*, i passar el valor generat com a argument del missatge `create()` enviat al component *objectiu*, és l'opció més raonable: no viola cap principi (a excepció, potser, d'*Alta cohesió*);[→] les visibilitats que exigeix concorden amb les visibilitats potenciades per *Creador orb*;[→] i l'acoblament de responsabilitats concorda amb l'acoblament semàntic d'aquestes, la qual cosa permet assegurar-ne la consistència.

8.2. Cohesió, pàgina 127

10.3. Principi *Creador orb*, pàgina 161

4 Generació per comprovació amb múltiples repositoris

4.1 Context de l'anàlisi

Múltiples repositoris. Considerem el cas en què l'*objectiu d'identificació* intervé en més d'un repositori. És a dir, el *repositori d'emmagatzematge* i el *repositori de contrast* poden no coincidir.

Reordatori sobre els àmbits d'identificació. Com hem vist,[→] l'ús d'un repositori centralitzat com a repositori de contrast proporciona identificació d'àmbit global i admet les migracions i les comparticions de realitzacions. Per contra, l'ús d'un fragment com a repositori de contrast proporciona identificació d'àmbit local i no admet ni les migracions ni les comparticions de realitzacions.

20.1.2. Generació per comprovació (GPC), pàgina 391

Mantenim la hipòtesi de treball. Per assegurar que la generació per comprovació es comporta correctament és indiferent si l'àmbit d'identificació que defineix el repositori de contrast és local o global: el que cal és assegurar que

mai no es generarà un valor existent en el repositori de contrast. Per guiar l'anàlisi considerem que el repositori de contrast és centralitzat i sense rèpliques.²³

Assumpció 62, pàgina 397

Experts en la generació per comprovació. Sota l'assumpció simplificadora que estem considerant, \rightarrow per ser expert en la generació per comprovació n'hi ha prou en tenir visibilitat del repositori de contrast.

4.2 Repositori d'emmagatzematge diferent del de contrast

? (68)

Fins a quin punt els resultats obtinguts sota la hipòtesi del repositori únic continuen essent vàlids sota la nova hipòtesi que el repositori de contrast i el d'emmagatzematge no coincideixen?

4.2.1 Individualització tardana

Violació de Creació complerta. La *individualització tardana* era desaconsellable perquè violava el principi de la *Creació complerta* (sota el supòsit de la *Identificació atributiva*). Però això és independent de si el repositori de contrast i el d'emmagatzematge coincideixen o no. El que impedeix el principi de la *Creació complerta* és el fet de postposar la creació del valor d'identificació fins el moment de l'emmagatzematge, independentment de si el repositori per crear el valor d'identificació coincideix amb el d'emmagatzemage o no.

Altres problemes amb múltiples repositoris. La *individualització tardana* significa crear el valor d'identificació en el moment d'inserir en un repositori, i per tant només té sentit si el repositori d'emmagatzematge (on inserim) coincideix amb el repositori de contrast (altrament el repositori d'emmagatzematge no és expert en la comprovació, i per tant el seu mètode d'inserció no pot rebre la responsabilitat de generar el valor d'identificació). És més, en cas de necessitat d'emmagatzematges múltiples caldria assegurar que el primer emmagatzematge es realitzés sempre sobre el repositori de contrast, amb l'objectiu de completar la realització amb el seu valor d'identificació. Tot plegat fa que la possibilitat de múltiples repositoris desaconselli fermament la *individualització tardana*.

4.2.2 Autoindividualització

L'*autoindividualització* era desaconsellable perquè violava el principi del *Desconeixement del grup* en exigir que l'objectiu tingués coneixement del repositori de contrast. Però això és independent del fet de si aquest repositori coincideix amb el d'emmagatzematge o no.

Assumpció 63, pàgina 399

²³Mantenim per tant l'assumpció de treball 63. \rightarrow

4.2.3 Individualització induïda

La *individualització induïda* era desaconsellable perquè violava el principi del *Baix acoblament*, però sobretot pels problemes de consistència que generava com a conseqüència d'haver desacoblat dues responsabilitats que semànticament estan estretament acoblades. Novament això és independent de la coincidència del repositori d'emmagatzematge amb el de contrast.

4.2.4 Individualització genètica

Necessitat del repositori de contrast. La *Individualització genètica* era el mecanisme aconsellable. Ara bé, per poder mantenir aquesta conclusió com a vàlida sota la nova hipòtesi de la no coincidència del repositori de contrast i del d'emmagatzematge, cal assegurar que el creador emprat és *expert* en la generació per comprovació. Però com hem vist, la condició necessària i suficient²⁴ per a aquesta expertesa és la de tenir visibilitat del repositori de contrast. Sota la hipòtesi del repositori únic l'expertesa era assegurada amb l'ús d'un *creador orb*; amb la nova hipòtesi les coses no són tant clares.

Els *creadors orbs* tenen visibilitat del repositori de contrast?

(69) ?

Visibilitats del creador orb. Suposem que el *creador orb* no tingui visibilitat del *repositori de contrast*. En aquest cas el *creador orb* no pot fer la generació per comprovació, i no hi ha més remei que la petició de creació rebuda incorpori el valor d'identificació. Però llavors estem en un cas d'*individualització induïda*, que sabem que és desaconsellable. Per tant cal admetre que el *creador orb* té visibilitat del *repositori de contrast*, i per tant és expert en la generació per comprovació.

4.3 Replicació i fragmentació del repositori de contrast

4.3.1 Necessitat d'emmagatzematge múltiple

Supòsit de l'anàlisi. Suposem ara que el repositori de contrast està replicat; i admetem que algunes de les rèpliques estiguin fragmentades.

Utilitat d'una rèplica centralitzada. La generació per comprovació del valor d'identificació s'ha de basar en tot el repositori de contrast, i per tant és aconsellable l'existència d'una rèplica centralitzada d'aquest que pugui ser usada pel mecanisme de comprovació.

²⁴Aquesta afirmació és conseqüència de l'assumpció simplificadora. → Sense ella la visibilitat del repositori de contrast és condició necessària, però no suficient, per a l'expertesa en la generació per comprovació.

Sincronització de repositoris. Cada nova realització introduïda en el repositori de contrast també s'ha d'introduir en totes les seves rèpliques, altrament les rèpliques deixarien de ser rèpliques. I això tant si el repositori de contrast és centralitzat com si és fragment d'una repositori distribuït.

Múltiple emmagatzematge. La sincronització de repositoris significa que cada necessitat d'emmagatzematge es tradueix en un múltiple emmagatzematge: com a mínim cal emmagatzemar en totes les rèpliques del repositori de contrast.

Altres emmagatzematges. A més, res impedeix que un cop creat un objecte i emmagatzemat en totes les rèpliques del repositori de contrast, aquest objecte s'emmagatzemi en d'altres repositoris, que no necessàriament han de ser fragments del repositori de contrast.

4.3.2 Responsable de l'emmagatzematge múltiple

? (70)

És aconsellable que qui emmagatzema en un repositori també es faci responsable de l'emmagatzematge en els altres?

Acoblament i cohesió dels emmagatzematges múltiples. Si bé el principi d'*Alta cohesió*[→] sembla aconsellar que un únic component rebi la responsabilitat de tots els emmagatzematges, el principi del *Baix acoblament*[→] ho desaconsella: l'*emmagatzemador* hauria d'estar acoblat a tot possible repositori, present o futur.

El *peticionari* com a *coordinador*. Una solució de compromís és demanar al *peticionari* de la creació que es preocupi de comunicar a tot aquell qui hi estigui interessat el fet que s'ha creat un nou objecte, per tal que qui en tingui responsabilitat d'emmagatzematge el pugui emmagatzemar en el repositori del qual és responsable. D'aquesta manera les responsabilitats dels diferents emmagatzemages estan distribuïdes per diferents components *emmagatzemadors*, i el *peticionari* actua com a *coordinador* de tots ells. El preu que cal pagar és l'*alt acoblament* del *peticionari*, que haurà de tenir coneixement dels diferents *emmagatzemadors*.

Excursió Mecanisme de publicació i subscripció. Una alternativa a donar al *peticionari* de la creació la responsabilitat de coordinar els diferents *emmagatzemadors* es basa en el mecanisme de publicació i subscripció. La idea és que els diferents *repositoris* es registrin com *subscriptors* de la creació d'una nova realització; i cada cop que es fa una nova creació el *creador* es preocupa de *publicar* aquest fet (és a dir, que avisa els subscriptors per tal aquests actuïn convenientment).

Per tal que la *publicació* pugui realitzar-se cal que tots els *creadors* tinguin visibilitat de l'estructura que emmagatzema els diferents subscriptors.

Una simplificació d'aquest requeriment és convertir una rèplica centralitzada del repositori de contrast, que anomenarem *repositori matriu*, en el responsable de la *publicació*. Llavors n'hi ha prou en assegurar que tot *creador* emmagatzemi com a mínim en el *repositori matriu*; exigir que tota rèplica del repositori de contrast es registri com a *subscriptora* del *repositori matriu*; i fer que l'emmagatzematge sobre el *repositori matriu* activi el mecanisme de *publicació*.

L'ús de llenguatges com el C++ o el Java que admeten propietats de classe, a part de les tradicionals propietats d'objecte, permet que els subscriptors es mantinguin amb visibilitat d'atribut des de la classe. En aquest cas les visibilitats exigides per la publicació estan assegurades, i no cal el *repositori matriu*.

5 Generació per comprovació: conclusions

Mecanisme d'individualització. La *individualització genètica* amb un *creador orb* apareix com el mecanisme més aconsellable, independentment de si el repositori d'emmagatzematge i el de contrast coincideixen, i independentment de si el repositori de contrast està replicat o fragmentat.

Funcionament del mecanisme. El *creador orb* és el responsable de la *creació* i l'emmagatzematge de l'objectiu. En rebre la petició de creació d'un nou objectiu el *creador orb* genera un nou valor d'identificació, i el passa com a argument al mètode `create()`.

Visibilitats. El *creador* tindrà visibilitat el *repositori d'emmagatzematge* (per definició de *creador orb*) i del *repositori de contrast* (per tal de fer la *comprovació de la identificació*).

6 Generació única

6.1 Plantejament de la generació única

6.1.1 Mecanismes per a la generació única

Generació única. Un mecanisme de *generació única* de valors d'identificació és un mecanisme que cada cop que s'usa genera un valor diferent; i per tant és independent de qualsevol repositori.

Mecanismes de generació única. Els mecanismes que s'usen per aconseguir la unicitat del valor generat són bàsicament dos: la generació aleatòria i la generació per seqüència infinita.

Generació per seqüència infinita. Disposem d'un mecanisme capaç de generar una seqüència infinita de valors no repetits. Per generar un nou valor d'identificació n'hi ha prou en generar un nou valor de la seqüència.

Exemple 93 L'enumeració de Σ^* és una seqüència infinita: la que conté tots els mots formats amb els símbols de Σ ; l'ordre de generació és l'ordre lexicogràfic. Coneixent el darrer valor generat es pot generar fàcilment el següent valor. Per exemple, sigui abb el darrer mot generat; i sigui $\Sigma = \{a, b\}$ l'alfabet considerat. Llavors el següent valor en l'ordre lexicogràfic és baa .

La seqüència dels nombres naturals és un altre mecanisme generador possible. N'hi ha prou en recordar el darrer valor generat per poder generar el següent.

Generació aleatòria. Disposem d'un mecanisme de generació aleatòria de valors tal que la probabilitat de generar un valor generat anteriorment és 0. Per tant la probabilitat que el valor generat sigui un identificador vàlid és 1.

Generació pseudo-aleatòria. La majoria dels mecanismes fiables per generar valors aleatoris són en realitat generadors *pseudo-aleatoris*: defineixen un algorisme més o menys elaborat de generació d'una seqüència infinita tal que la probabilitat que un mateix valor aparegui dos cops en la seqüència és tant petit com es vulgui. Així a la pràctica la generació aleatòria es pot veure com un cas particular de la generació per seqüència infinita; la diferència essencial és que hi ha una determinada probabilitat (que tendeix a zero) que la seqüència pugui tenir valors repetits.

6.1.2 Estat intern de la generació

Estat de la generació. Sigui quin sigui el mecanisme de la generació única, en general es tracta d'un mecanisme amb *estat*. Potser caldrà recordar els darrers paràmetres de generació emprats; potser caldrà recordar el darrer valor generat; o potser caldrà recordar quants valors s'han generat. Sigui com sigui, cal recordar l'*estat* intern del mecanisme en el moment actual de la generació.

Experts en la generació única. Els experts en la generació única són els components que tenen coneixement de l'estat de la generació.

Àmbit d'identificació

Extensió notacional. Fent un abús de la nomenclatura direm que l'*àmbit d'identificació* d'un component C és el component I , per expressar que l'*àmbit d'identificació* de C és el d'un repositori visible des d'un objecte $i : I$.

Àmbit d'identificació. Tots els components que coneixen el mateix estat de la generació comparteixen l'*àmbit d'identificació*. Si aquests components usen repositoris diferents, l'*àmbit d'identificació* serà la unió de tots aquests repositoris.

Exemple 94 Suposem un mecanisme de generació única per les *Inscripcions*. Suposem que *Caminada* n'és la creadora, i que el repositori de *Inscripcions* està fragmentat per *Caminades*.

Si cada *Caminada* manté el seu propi estat del mecanisme de generació, llavors l'*àmbit d'identificació* de *Inscripció* és el repositori mantingut per la $c : \text{Caminada}$ pertinent. Amb l'abús notacional

exposat més amunt direm que l'àmbit d'identificació d'Inscripció és la Caminada. En temres planers el que estem dient és que les inscripcions no es poden repetir dins una mateixa c: Caminada.

Si totes les Caminades comparteixen un mateix estat del mecanisme de generació, llavors l'àmbit d'identificació d'Inscripció és la unió de tots els repositoris d'Inscripció presents en les diferents Caminades; és a dir, l'àmbit d'identificació d'Inscripció és el repositori fragmentat per Caminades. Amb l'abús notacional exposat més amunt, l'àmbit d'identificació és el de l'objecte que té visibilitat de totes les Caminades; que en el nostre cas és el controlador. Però com que el controlador representa tot el sistema o subsistema considerat, podem afirmar que l'àmbit d'identificació d'Inscripció és l'àmbit global.

6.2 Individualització tardana

Es mantenen els contres. Com hem vist, el principi de la *Creació complerta* (sota la hipòtesi de la *Identificació atribuïtiva*) desaconsella postposar la individualització fins en el moment de fer la inserció en un repositori; i els problemes de consistència davant la possibilitat de múltiples repositoris refermen encara més el rebuig a la *Individualització tardana*.

Visibilitats exigides. Si malgrat tots els problemes exposats encara volguéssim usar la *individualització tardana* caldria, pel principi *Expert*, assegurar que tot repositori responsable de la individualització tingués visibilitat de l'estat de la generació.²⁵

6.3 Autoindividualització

Desapareix un inconvenient. El principi del *Desconeixement del grup* ara no és aplicable: el mecanisme de generació és independent del repositori o del grup.

Arguments de `create()`. Així podria ser el disseny de `create()` qui generés l'identificador. Però per fer-ho cal passar-li com a argument l'estat de la generació.²⁶

Acoblament i cohesió. El resultat és que baixa la cohesió del mètode `create()` (crea dos elements: un objecte i el seu identificador),²⁷ i s'acobla el component al mecanisme de generació del valor d'identificació. I això pot ser un problema: el canvi en el mecanisme de generació obliga a canviar el component *objectiu*. És a dir, l'*autoindividualització* acobla excessivament dues responsabilitats (la

²⁵De fet hem substituït la necessitat de la visibilitat del repositori de contrast per la visibilitat de l'estat de la generació.

²⁶De fet cal donar coneixement al mètode de creació tant de l'estat de la generació com del procediment de generació a partir de l'estat.

²⁷Com hem vist hi ha arguments per defensar que aquí no hi ha pèrdua de cohesió. Recordem que la cohesió és una mesura qualitativa, i per tant subjectiva.

individualització i la creació d'una nova realització)²⁸ que semànticament no necessàriament ho han d'estar.

6.4 Individualització induïda

Anàlisi anterior no aplicable. El desacoblament excessiu de les responsabilitats d'individualització i creació desaconsellaven la *individualització induïda*, per mor de la facilitat de caure en inconsistències d'identificació. Cal observar però com aquestes inconsistències eren causades per la possibilitat que l'*individualitzador* generés valors d'identificació repetits. Però si el mecanisme de generació dels valors d'identificació és la *generació única* aquest problema desapareix.

El llarg viatge dels identificadors. El *peticionari* demana a l'*individualitzador* un valor d'identificació; i tot seguit el passa al *creador* per tal que creï una nova realització identificada amb el valor donat. Així, el valor d'identificació creat viatja, en forma de valor de retorn, al llarg de tota la cadena de delegació que va des del *peticionari* a l'*individualitzador*; i, en forma d'argument de la petició de creació, al llarg de tota la cadena de delegació que va des de l'*individualitzador* al *creador*. En determinades circumstàncies aquest flux es pot considerar que viola el principi de la *Cistella*.

6.5 Individualització genètica

Esperresa perduda. El *creador* de l'objectiu no és pas *expert* en la *generació única*. Ni tant sols si es tracta d'un *creador orb*: la visibilitat del *repositori d'emmagatzematge* és del tot independent a la visibilitat sobre l'*estat de la generació*.

Mantenim les conclusions. Malgrat tot, adoptar el mecanisme de la *individualització genètica* (forçant les visibilitats que calgui), simplifica els dissenys: independentment de si el mecanisme de generació és per comprovació o per generació única, seguim les mateixes pautes de disseny per a la individualització.

7 Principi de la *Individualització efectiva*

Principi



Individualització efectiva. Per tal de generar valors d'identificació emprem la *individualització genètica*; és a dir, donem la responsabilitat de crear el valor d'identificació al creador de l'objectiu que volem individualitzar.

²⁸De fet converteix la responsabilitat d'individualització en part de la responsabilitat de creació. La *individualització genètica* també assigna ambdues responsabilitats a un mateix component, però cadascuna d'elles és independent a l'altra. En un casament jeràrquic o de supeditació; en l'altre l'acoblament és circumstancial o de coincidència en pla d'igualtat en un mateix component.

Aquest principi intenta resumir en quatre mots tota l'anàlisi i exposició sobre la generació dels valors d'identificació. Cal tenir present però que el principi de la *Individualització genètica* és defensable si l'usem simultàniament al principi del *Creador orb*; altrament les visibilitats necessàries potser no estaran assegurades, i en forçar-les potser es produeixen acoblaments innecessaris.

Principis i definicions del capítol

Principis

Creació complerta, 400

Desconeixement del grup, 404

Individualització efectiva, 416

Definicions

Repositori de contrast, 396

Capítol 21

Identificació: conclusions

1	Identificadors: conclusions	421
1.1	Problemes comunicatius	421
1.2	Ús dels identificadors	422
1.3	Identificació interna	423
2	^{GLS} La identificació d'Inscripcions	424

Contingut detallat del capítol 21

1	Identificadors: conclusions	421
	Una anàlisi complexa	421
	Objectiu de les respostes	421
1.1	Problemes comunicatius	421
1.2	Ús dels identificadors	422
1.3	Identificació interna	423
2	^{GLS} La identificació d'Inscripcions	424
	Solució proposada	424
	Correctesa de la proposta	424
	Fragmentació del repositori	424
	Comprovació d'identificador	424

1 Identificadors: conclusions

Una anàlisi complexa. Com hem vist la problemàtica dels identificadors és força complexa. Però després de l'anàlisi realitzada ja estem en condicions de respondre les qüestions que ens preocupen.

Objectiu de les respostes. Les respostes que segueixen tenen en compte els consells dels diferents principis de disseny, la facilitat del disseny resultant (en defugir possible complicacions) i el denominador comú de totes les possibles situacions (i d'aquesta manera podem dissenyar sense pensar, per exemple, en quin és exactament el mecanisme de generació d'identificadors que s'emprarà).

1.1 Problemes comunicatius

Com podem comunicar el sistema amb el seu exterior? (71) ?

Comunicació de noms.

La comunicació es fa a través d'un llenguatge de noms. Per tant només podem comunicar allò que tingui nom.

(93)

Quins són els elements bàsics de la comunicació? (72) ?

Elements comunicables.

Comuniquem conceptes, components i les seves realitzacions.

(94)

Tot component és vàlid per a la comunicació? (73) ?

Components comunicables.

Els components que volguem comunicar han de ser especulars i seguir el principi de la *Franquícia obligada*. A més el component, com a model, ha de seguir el principi de l'*Espill* i el de la *Franquícia obligada*. (*Component fortament especular*)

(95)

? (74) Tot objecte és vàlid per a la comunicació?

(96)

Objectes comunicables.

Només es poden comunicar objectes que siguin realització d'un component comunicable

? (75) Tot conjunt de noms és vàlid com a llenguatge de comunicació?

(97)

Com a llenguatge de comunicació cal usar les propietats d'identificació sobre els components comunicable o sobre un subconjunt @@@
(Propietat d'identificació especular)

1.2 Ús dels identificadors

? (76) Quines condicions imposa al sistema l'aparició d'un identificador com a argument d'un ES?

(98)

Els identificadors modifiquen la interfície del sistema.

El sistema ha de proporcionar al seu client un mecanisme de comprovació d'identificador. (Principi de la *Identificació robusta*)

? (77) Quines condicions imposa en el disseny d'un ES l'aparició d'un identificador com a argument de l'ES en qüestió?

(99)

Els identificadors imposen responsabilitats al disseny.

Si estem en un cas d'ús de la referència, una de les primeres responsabilitats ha de resoldre és la recuperació de la realització. (Principi de l'*Encaixada*)

Quines condicions de visibilitat imposa l'aparició d'un identificador com a argument d'un ES?

(78) ?

Els identificadors exigeixen visibilitats.

(100)

El controlador ha de tenir visibilitat (directa o indirecta) de l'àmbit d'identificació corresponent. (Principi dels Àmbits accessibles)

Com decidim quin àmbit d'identificació és el correcte?

(79) ?

Els identificadors no han de violar *Espill*.

(101)

L'àmbit d'identificació no ha d'imposar noves associacions a MC. (Principi de la *Consistència individualitzadora*)

1.3 Identificació interna

Qui és el responsable de generar (crear) els valors d'un identificador?
Qui és el *generador d'identificadors*?

(80) ?

Responsable de generar l'identificador.

(102)

El *generador d'identificadors* és el *creador*¹ de l'objecte que es vol identificar. (Principi de la *Individualització efectiva*)

Què cal conèixer per crear un valor únic per un identificador?

(81) ?

¹No cal especificar el tipus de creador. Els principis que ens aconsellen sobre la responsabilitat de creació ja ens demanen en primera instància l'ús creadors canònics, que són una restricció dels creadors orbs.

(103)

Expertesa per a la generació de l'identificador.

En el cas de la *generació per comprovació* cal conèixer el *repositori de contrast*. En el cas de la *generació única* cal conèixer l'*estat* concret del mecanisme de generació.

@@@ La prefgutna i la resposta estan repes, no????

? (82)

Qui és el responsable de crear el valor d'identificació d'un objecte?

(104)

Emprem la *individualització genètica*; és a dir, donem la responsabilitat de crear el valor d'identificació al creador de l'objectiu que volem individualitzar.

2 ^{GLS} La identificació d'Inscripcions

Solució proposada. Per tal de resoldre el problema de la identificació de les Inscripcions usem la següent solució de disseny

- Identificació: interna
- Àmbit d'identificació: la Caminada

🔗 (83)

La Inscripció és d'identificació interna, i àmbit d'identificació la Caminada

Correctesa de la proposta. Com hem vist la Caminada és un creador canònic; per tant es compleixen tant els principis de la *Consistència individualitzadora* (l'àmbit d'identificació es pot extreure de MC) com el de la *Individualització efectiva* (el creador és l'individualitzador).

Fragmentació del repositori. La solució proposada significa que les Inscripcions s'emmagatzemen en un repositori fragmentat per Caminades; i per tant el codi d'una Inscripció es pot repetir entre Caminades diferents.

Comprovació d'identificador. En els ES analitzats fins ara no apareix el codi d'Inscripció com a argument. Per tant no cal oferir cap mecanisme de comprovació d'identificador.

Capítol 22

GLS Contracte de l'ES *nouParticipant* (nom, edat)

1	Contracte de partida	428
2	Àmbits d'identificació	429
2.1	Requeriments i àmbit d'identificació	429
2.2	La Caminada com a àmbit d'identificació del <i>Partícep</i>	430
2.3	La Inscripció com a àmbit d'identificació del <i>Partícep</i>	432
2.4	El concepte <i>Partícep</i> amb identificació interna	434
2.5	Conclusions sobre l'àmbit d'identificació	435
3	Desenvolupament àgil	436
3.1	Completar les especificacions des del disseny	436
3.2	Dissenyar amb especificacions incomplertes	437
4	Ús de la <i>Presentació</i>	437
5	Conclusions sobre el contracte	439
6	Retorn a l'especificació	440

Contingut detallat del capítol 22

1	Contracte de partida	428
	El concepte <i>Partícep</i>	428
	Utilitat del concepte <i>Partícep</i>	428
	Objectiu del capítol	428
	Significat de la pregunta	428
	Motivació del capítol	428
2	Àmbits d'identificació	429
2.1	Requeriments i àmbit d'identificació	429
	Requeriments que cal complir	429
	Identificació externa	429
	Refinament de la PRE	429
	PRE massa forta	429
	Afebliment de la PRE	429
	El recurs de l'àmbit de la identificació	429
	El salt al disseny	429
	Possibles àmbits d'identificació del <i>Partícep</i>	430
2.2	La Caminada com a àmbit d'identificació del <i>Partícep</i>	430
2.2.1	Refinament del contracte	430
	Límits de la proposta	430
	Revisió del contracte	430
	Conseqüències en el disseny	430
2.2.2	Comprovació d'identificador	431
	Necessitat de la comprovació d'identificador	431
	Àmbit implícit d'identificació	431
	El significat de <i>participa</i>	431
	Necessitat de refinament del significat de <i>participa</i>	431
	Associacions presents a MC	431
	Descomposició de <i>participa</i>	431
	Àmbit explícit d'identificació	431
	Comunicació del 'àmbit d'identificació	431
2.3	La Inscripció com a àmbit d'identificació del <i>Partícep</i>	432
	Utilitat de l'anàlisi	432
	Semàntica de la proposta	432
	Coincidència d'identificadors i coincidència de realitzacions	432
	Compartició de realitzacions	432
	Hem equivocat la mirada	432
	Rectifiquem cap el bon camí	433
	Unicitat del nom de <i>persona</i>	433
	Incompletesa de la proposta	433
2.4	El concepte <i>Partícep</i> amb identificació interna	434
	Una nova proposta	434
	Validesa de la proposta	434
	Conseqüències de la identificació interna	434

Llibertat excessiva?	434
La relació entre <i>Persona</i> i <i>Particip</i>	435
Propietat d'identificació errònia	435
Llibertat necessària	435
2.5 Conclusions sobre l'àmbit d'identificació	435
Propostes de l'anàlisi	435
El filtre del disseny	435
El filtre de la simplicitat	435
3 Desenvolupament àgil	436
3.1 Completar les especificacions des del disseny	436
Especificació incompleta	436
Revisió de l'especificació des del disseny	436
Desenvolupament àgil	436
3.2 Dissenyar amb especificacions incomplertes	437
Desenvolupament àgil revisat	437
Mans a l'obra	437
Identificació global	437
Àmbits d'identificació locals possibles	437
Tipus d'identificació	437
Del disseny a l'especificació	437
4 Ús de la Presentació	437
<i>Domini</i> i <i>Presentació</i>	437
Objectius en el disseny del <i>Domini</i>	438
Objectius en el disseny de la <i>Presentació</i>	438
La <i>Presentació</i> i els <i>Participes</i>	439
5 Conclusions sobre el contracte	439
Propostes de l'anàlisi	439
Repercussions en el contracte	439
6 Retorn a l'especificació	440
Ús d'especificacions incomplertes	440
Desenvolupament en cascada	440
Desenvolupament àgil en cascada	440
Especificació i disseny	440
Restricció del desenvolupament àgil en cascada	440
Desenvolupament àgil en cascada de <i>nouParticipant (nom, edat)</i>	440
Parèntesi expositiu	440

1 Contracte de partida

Contracte <i>nouParticipant</i> (<i>nom</i> , <i>edat</i>)	
Paràmetres	
• <i>nom</i> :	Nom del participant
• <i>edat</i> :	Edat del participant
PRE	
1.	Hi ha una <i>i</i> :Inscripció activa
2.	El participant és nou
POST	
1.	Hi ha una nova realització <i>p</i> :Partícep
2.	Hi ha un enllaç entre <i>i</i> i <i>p</i> , corresponent a l'associació inscrit en
3.	La inscripció <i>i</i> es manté activa

24. ^{GLS}La idea de participació, pàgina 471

El concepte Partícep. Bé sigui per intuïció, per anàlisi sistemàtica,^{1,→} o per moltes hores d'experiència, a MC enlloc d'usar el concepte *Persona* fem aparèixer el concepte *Partícep*.² La idea és que per una mateixa *p*:*Persona* hi ha tantes realitzacions del concepte *Partícep* com cops la *p*:*Persona* apareix en el sistema.

Utilitat del concepte Partícep. L'ús del concepte *Partícep* simplifica molt tant l'especificació com el disseny: el *p*:*Partícep* està lligat a una sola *i*:*Inscripció*, i per tant a una sola *c*:*Caminada*, i d'aquesta manera totes les associacions es simplifiquen.

Objectiu del capítol. L'objectiu d'aquest capítol és respondre la pregunta següent:

? (84)

Què vol dir que el que expressa la PRE en demanar que el *participant* sigui nou?

Significat de la pregunta. La pregunta acabada de plantejar pot semblar absurda: si cada cop que una *Persona* intervé en el sistema considerem que tenim un nou *Partícep*, no és possible obtenir *Partíceps* repetits. Malgrat tot el concepte *Partícep* ha aparegut per simplificar el model; mentre que el contracte fa referència a la idea de *participant*. El que cal doncs respondre és: què és un *participant*? Com es pot expressar en termes del concepte *Partícep*?

¹En el capítol 24. ^{GLS}La idea de participació, pàgina 471, es realitza l'anàlisi sistemàtica.

²El nom del concepte no és massa afortunat, però s'ha escollit per no interferir en la nomenclatura, força més afortunada, seguida en el capítol 24. ^{GLS}La idea de participació, pàgina 471.

Motivació del capítol. El que perseguim en aquest capítol és mostrar els tipus de raonaments que s'han de fer quan es parteix d'una especificació potser no suficientment madurada. En el capítol 24.^{GLS} *La idea de participació*, pàgina 471[→] repetim l'anàlisi però a partir d'una construcció sistemàtica del MC.

En aquest capítol intentem millorar el contracte de l'ES **des del disseny**

24.^{GLS} *La idea de participació*, pàgina 471

2 Àmbits d'identificació

2.1 Requeriments i àmbit d'identificació

Requeriments que cal complir. L'enunciat exigeix que una mateixa persona:

1. No pugui aparèixer dos cops en la mateixa inscripció
2. No pugui aparèixer en inscripcions diferents sobre la mateixa caminada
3. Es pugui inscriure en caminades diferents
4. Si s'inscriu en caminades diferents aquestes no es poden realitzar en la mateixa data.

Identificació externa. L'ES *nouParticipant (nom, edat)* rep com a argument el *nom d'una persona*. Si usem aque *nom de persona* com a identificador de *Partícep*, estem en un context d'*identificació externa*, i d'aquí la necessitat d'una PRE que assegurí les propietats d'identificació. Aquest és justament el paper de la PRE "El participant és nou".

Refinament de la PRE. Estem ja en condicions de fer un primer refinament de la PRE que estem analitzant: cal exigir que no hi hagi cap *Partícep* amb el mateix *nom*.

PRE massa forta. En una primera ullada podem observar com la PRE acabada d'exposar és massa forta: en no admetre *Partíceps* diferents amb el mateix *nom*, no admetem que que una mateixa *persona* pugui aparèixer dos cops en el sistema. I per tant no admetem que una mateixa *persona* pugui participar en més d'una *Caminada*.

Afebliment de la PRE. Per afeblir la PRE podem exigir que la prohibició dels *Partícep* amb el mateix *nom* es redueixi a un context determinat. És a dir, el *nom de persona* individualitza els *Partíceps* dins de determinat context.

El recurs de l'àmbit de la identificació. En l'especificació no hem introduït cap mecanisme per reduir el context d'identificació. En canvi sí que l'hem introduït en el disseny: es tracta de l'*àmbit de la identificació*, que hem definit com el repositori on s'ha de complir la unicitat de la propietat d'identificació.[→] Anem doncs al disseny i analitzem els possibles *l'àmbits de la identificació*.

19.4.1. Àmbit d'identificació, pàgina 375

El salt al disseny. Aquest salt al disseny per resoldre llacunes de l'especificació el justifiquem a l'apartat 22.3. [Desenvolupament àgil](#), pàgina 436. → L'alternativa és aprofundir en l'especificació, tal com fem al capítol 24. ^{GLS}[La idea de participació](#), pàgina 471. →

22.3. [Desenvolupament àgil](#), pàgina 436

24. ^{GLS}[La idea de participació](#), pàgina 471

Possibles àmbits d'identificació del Partícep. Si usem àmbit global impedim que una mateixa persona s'inscrigui a caminades diferents (no complim l'exigència 3); si usem àmbit local a la *c*:Caminada aconseguim les exigències 1, 2 i 3; si usem àmbit local a la *i*:Inscripció aconseguim l'exigència 1 i 3.

2.2 La Caminada com a àmbit d'identificació del Partícep

2.2.1 Refinament del contracte

Límits de la proposta. A partir dels requeriments sembla que l'àmbit d'identificació correcte és la Caminada: un *p*:Partícep no és repetible dins d'una mateixa *c*:Caminada, però res no impedeix que es repeteixi en Caminades diferents. Queda, però, per resoldre el tema de la no coincidència de data.

Revisió del contracte. Si usem la Caminada com a àmbit de la identificació caldrà una PRE en la introducció de nous participants que assegurí que el nou nom és inexistent en la Caminada.³ A més, com que apareix un identificador (el *nom*) com a argument d'un ES, caldrà oferir algun mecanisme de *comprovació d'identificador*.⁴

Contracte (Unicitat :Caminada)	
Paràmetres	
• <i>nom</i> :	Nom del participant
• <i>edat</i> :	Edat del participant
PRE	
1.	Hi ha una <i>i</i> :Inscripció activa
POST	
1.	Hi ha una nova realització <i>p</i> :Partícep
2.	Hi ha un enllaç entre <i>i</i> i <i>p</i> , corresponent a l'associació inscrit en
3.	La inscripció <i>i</i> es manté activa

Conseqüències en el disseny. La segona PRE de *nouParticipant ()* en el disseny és fàcilment expressable en termes d'àmbits d'identificació:

El nom és un identificador de Partícep, dins l'àmbit de la *c*:Caminada associada a la *i*:Inscripció activa.

³No es tracta d'una PRE nova. Simplement és el refinament de "El participant és nou".

⁴El mecanisme de comprovació introduït l'analitzem a 22.2.2.2. [Comprovació d'identificador](#), pàgina 431. →

2.2.2 Comprovació d'identificador

Necessitat de la comprovació d'identificador. En l'ES tenim un argument que és un identificador de `Partícep` dins l'àmbit d'identificació d'una `:Caminada`. El principi `@@` \rightarrow demana l'existència d'un mecanisme per a la comprovació d'aquest identificador.

Contracte [bool]
<p>existeixParticipant(nom), colback=white, halign title=flush center ,options@for=contracte] Paràmetres</p> <ul style="list-style-type: none">• <i>nom</i>: Nom del participant <p>PRE</p> <ol style="list-style-type: none">1. Hi ha una <code>:Caminada</code> activa <p>POST</p> <ol style="list-style-type: none">1. Cert \Leftrightarrow A la <code>:Caminada</code> activa hi participa un <code>p:Partícep</code> amb <code>p.nom=nom</code>

???, pàgina ??

Àmbit implícit d'identificació. El `nom` és un identificador (en un àmbit d'identificació determinat). La comprovació d'identificador presentada considera l'àmbit d'identificació predefinit; i d'aquí la PRE.

El significat de *participa*. En la POST *participa* és l'associació que lliga *Caminada* amb *Partícep*.

Necessitat de refinament del significat de *participa*. És ben possible que en el MC aquesta associació no existeixi. En aquest caldrà expressar el significat de *participa* en termes del model.

Associacions presents a MC. Les associacions que segur que té el nostre MC són:

- *referent a:Caminada-Inscripció*
- *inscrit:Inscripció-Partícep*

Descomposició de *participa*. La semàntica del *participa* es pot expressar com la composició de *referent a* i *inscrit*:

$$participa(x, y) = inscrit(x, z) \bullet referent_a(z, x)$$

Àmbit explícit d'identificació. Alternativament, enlloc de pressuposar un àmbit d'identificació conegut, podem donar aquest àmbit com a argument de l'ES.

Comunicació del 'àmbit d'identificació. Un àmbit d'identificació és un repositori, i per tant queda definit per un objecte (l'origen de l'explosió) i una expansió. Si assumim que l'expansió és implícita llavors el que cal és comunicar l'objecte origen de l'explosió. I per fer-hi hi ha dues possibilitats: comuniquem l'objecte, o en comuniquem el seu identificador. D'aquí que tenim les següents possibilitats:

- *existeixPartícep*(*nomCaminada*, *nomParticipant*):<bool>
- *existeixPartícep*(*c:Caminada*,*nomParticipant*, *nomParticipant*):<bool>

I aquest àmbit el podem comunicar amb l'objecte que

2.3 La Inscripció com a àmbit d'identificació del Partícep

Utilitat de l'anàlisi. Tot seguit analitzem la proposta d'usar la Inscripció com a àmbit d'identificació de Partícep. Es tracta d'una proposta que inicialment hem descartat per no complir els requeriments. La seva anàlisi, però, ens mostra unes subtilitats interessants.

Semàntica de la proposta. La proposta es basa en mantenir el *nom* com a identificador generat externament al sistema, i usar la Inscripció com a àmbit d'identificació. Llavors un mateix *nom* no pot estar repetit dins d'una mateixa Inscripció (complim el requeriment 1), però sí que pot aparèixer en Inscripcions diferents de la mateixa Caminada (violem el requeriment 2), o de Caminades diferents (complim el requeriment 3).

Coincidència d'identificadors i coincidència de realitzacions. Els identificadors s'usen per individualitzar els objectes del seu col·lectiu; i per tant hom espera que en usar el mateix identificador en moments diferents hom obtindrà el mateix objecte.⁵ Cal tenir present, però, que aquesta afirmació només és certa si mantenim l'àmbit d'identificació: en general, *un mateix identificador, en àmbits d'identificació diferents, es refereix a objectes diferents*.⁶

Compartició de realitzacions. L'ús de l'àmbit d'Identificació, com àmbit d'identificació del Partícep, amb identificació externa (el *nom*), pot permetre l'aparició del mateix Partícep en Inscripcions o Caminades diferents. Però per tal que això sigui possible caldrà introduir mecanismes per poder compartir realitzacions de Partícep en diferents repositoris.

Hem equivocat la mirada. Cal introduir mecanismes de compartició de realitzacions de Partícep?⁷ La semàntica del concepte *Partícep* és que tenim

⁵Això és així sempre i siguem fidels al principi de l'*Estabilitat d'identificació*; vegeu l'apartat 5, pàgina 380.

⁶Si un àmbit d'identificació és un subconjunt d'un altre, totes les realitzacions de l'àmbit restringit també ho són de l'àmbit extès. Llavors un mateix identificador usat en àmbits diferents identifica un sol objecte. Aquesta però no és la situació habitual.

⁷Pensem una mica en el que això significa: per introduir un nou *participant* caldria decidir si es tracta d'un de nou *Partícep*, o si per contra cal cercar un *p:Partícep* ja existent en el sistema.

22.2 Àmbits d'identificació

J.M. Merenciano

una nova realització del concepte cada cop que una mateixa *Persona* apareix en el sistema. Per tant, quin sentit té la compartició de *partíceps*? Cap ni una. Hem enfocat l'anàlisi cap el lloc incorrecte.

Rectifiquem cap el bon camí. Les unicitats que estem cercant no han de fer referència a *Partícep*, que sabem que en tenim un de nou per cada aparició d'una mateixa *Persona*, sinó que han de fer referència a *Persona*. Però d'aquesta, en el nostre MC només en queda el *nom*.

Unicitat del nom de persona. L'ús de l'àmbit d'Identificació, com àmbit d'identificació del *Partícep*, amb identificació externa (el *nom*), permet que *Partíceps* diferents però amb el mateix *nom* puguin aparèixer en *Inscripcions* o *Caminades* diferents. Per tant podem reafirmar la idea inicial que la proposta viola el requeriment 2 i compleix el requeriment 3.

Incompletesa de la proposta. Malgrat tot la proposta no és errònia; simplement és incompleta: cal afegir les condicions per evitar la repetició d'un mateix *nom* en *Inscripcions* diferents d'una mateixa *c:Caminada*. Un altre tema, però, és com es completa la proposta.

Contracte *nouParticipant (nom, edat)* (Amb el concepte *Partícep*^a)

^aManca assegurar el requeriment que *Inscripcions* diferents d'una mateixa *c:Caminada* no poden compartir *nom*.

Paràmetres

- *nom*: Nom del participant
- *edat*: Edat del participant

PRE

1. Hi ha una *i:Inscripció* activa
2. La *i:Inscripció* activa no conté cap *p:Partícep* amb *p.nom = nom*^a

POST

1. S'ha creat una nova realització *p:Partícep*
2. S'ha creat un enllaç entre *i* i *p*, corresponent a l'associació *inscrit* en
3. La inscripció *i* es manté activa

^aMés abreujadament: El *nom* és un identificador dins l'àmbit de la *i:Inscripció*.

Contracte [bool]

existeixParticipant(nom), *colback=white*, *halign title=flush center*, *options@for=contracte*] Paràmetres

- *nom*: Nom del participant

PRE

1. Cert

POST

1. Cert \Leftrightarrow la inscripció activa veu un *p:Partícep* amb *p.nom=nom*⁸

2.4 El concepte *Partícep* amb identificació interna

Una nova proposta. Una altra possibilitat és considerar que el *nom* no és identificador. Assumim que el sistema no s'ha de preocupar de les *persones*, sinó només de les *participacions*. Cada *participació* és una de nova, i al sistema no li interessa si fa referència a una persona coneguda o no; ni tant sols si es tracta d'una persona amb d'altres participacions. És a dir, usem *identificació interna*.

Validesa de la proposta. Amb la *identificació interna* sembla que ens apropem més a la idea inicial del concepte *Partícep*: cada cop que una *Persona* s'acosta al sistema tenim un nou *Partícep*; per tant el *nom* de la *Persona* no en pot ser l'identificador.⁹

Conseqüències de la identificació interna. La *identificació interna* significa que el *nom* rebut com a argument de l'ES és un atribut del nou *Partícep*. A més, en no ser identificador, no hi ha cap condició restrictiva sobre el *nom* (tots els valors donats són vàlids); i desapareix la necessitat de la *comprovació d'identificador*.

Contracte (Amb el concepte *Partícep* i identificació interna)

Paràmetres

- *nom*: Nom del participant
- *edat*: Edat del participant

PRE

1. Hi ha una *i:Inscripció* activa

POST

1. S'ha creat una nova realització *p:Partícep*
2. S'ha creat un enllaç entre *i* i *p*, corresponent a l'associació *s'inscriu*
3. La inscripció *i* es manté activa

⁸Hem aprofitat l'existència d'una *Inscripció* activa. Alternativament haguéssim pogut plantejar la *comprovació d'identificador* de la següent manera, per exemple: *existeixPartícep(i:Inscripció, nomParticipant):<bool>*.

⁹Abans hem intentat compaginar la semàntica del concepte *Partícep* amb la identificació a través del *nom de la persona* mitjançant l'ús d'àmbits d'identificació. Com veurem al capítol 24. ^{GLS}La idea de *participació*, pàgina 471, aquest recurs de fet implica un canvi en el MC. Aquí el que es proposa és usar la semàntica del concepte *Partícep* en totes les seves conseqüències; i en concret, en no necessitar el recurs de l'àmbit d'identificació, mantenint el MC.

Llibertat excessiva?.. L'ús de la *identificació interna* deixa total llibertat als *noms*: ni hi posem restriccions, ni oferim mecanismes de comprovació. És excessiva aquesta llibertat? Com veurem tot seguit, de fet soluciona un problema que de moment havíem bandejat.

La relació entre *Persona* i *Partícep*. Com sabem quan dos *Partíceps* diferents modelen la mateixa *Persona*? La coincidència de *noms* entre dos *Partíceps* no és suficient: fins i tot usant *noms* i *cognoms* hi pot haver duplicitats. De fet pel sistema és impossible determinar si dues *Persones* (o dos *Partíceps*) amb el mateix *nom* són la mateixa *Persona* real o no.

Propietat d'identificació errònia. La problemàtica de la dificultat d'individualitzar una *p:Persona* a través del seu *nom* és conseqüència d'usar com a *identificador* una propietat que no és una *propietat d'identificació*: no es compleix la condició d'unicitat. El *nom* no és un *identificador* de *Persona*.

Llibertat necessària. La no restricció en els *noms* que significa la *identificació interna* dels *Partíceps* soluciona la problemàtica de la no unicitat del *noms* de les *Persones*. Per tant la llibertat que implica la proposta, lluny de ser excessiva, pot ser considerada necessària.

2.5 Conclusions sobre l'àmbit d'identificació

Propostes de l'anàlisi. Les solucions possibles són:

- Identificació externa; àmbit d'identificació la *Caminada*.
 - Assegura tots els requeriments
 - Assumeix que el *nom* identifica les *persones* (siguin aquestes o no un concepte de MC).
- Identificació externa; àmbit d'identificació la *Inscripció*.
 - Cal afegir mecanismes i/o condicions per assegurar que *Partíceps* amb el mateix *nom* no apareixen en *Inscripcions* diferents de la mateixa *c: Caminada*
 - Assumeix que el *nom* identifica les *persones* (siguin aquestes o no un concepte de MC).
- Identificació interna
 - Cal afegir mecanismes per reintroduir les restriccions exigides en els requeriments.
 - No assumeix que el *nom* identifica les *persones* (siguin aquestes o no un concepte de MC).

El filtre del disseny. De les propostes de l'anàlisi, la primera viola el principi de la *Consistència identificadora*: a MC no hi ha una associació entre *Caminada* i *Partícep*; en canvi sí que n'hi ha entre *Inscripció* i *Partícep*. Per tant la solució que caldrà adoptar és la que considera la *Inscripció* com a àmbit d'identificació, o bé la que considera l'àmbit global.

El filtre de la simplicitat. En cas d'usar l'àmbit d'identificació la *Inscripció* i identificació externa caldrà completar l'especificació. A més haurem de conviure amb un identificador, el *nom* que no està clar que compleixi la propietat d'unicitat exigida a tota propietat d'identificació. Per contra, si usem la identificació interna tan el MC com el disseny semblen més simples; el preu és pagar és la llibertat sobre els *noms*, que pot portar a violar els requeriments. Malgrat tot, i amb tota la cautela que calgui, de moment ens decantem per la simplicitat.

➤ (86)

Useu la o bé *Inscripció* com a àmbit d'identificació del *Partícep*, o bé àmbit global

➤ (87)

Useu identificació interna del *Partícep* (Aquesta decisió és presa amb cautela i potser caldrà reconsiderar-la)

3 Desenvolupament àgil

3.1 Completar les especificacions des del disseny

Especificació incompleta. En l'especificació hem decidit que el concepte *Persona* no era el més adient: complicava excessivament tant el MC com els diferents CU. I hem introduït el concepte *Partícep*, que simplifica molt les coses.¹⁰ El resultat però deixava una llacuna important en el contracte de l'ES *nouParticipant* (*nom*, *edat*), en concret, el significat de l'enunciat "el *participant* és nou".

Revisió de l'especificació des del disseny. A partir d'awuï hem saltat al disseny. Un cop immersos en ell, i amb molts més elements a la mà, hem intentat resoldre les llacunes de l'especificació. Per ser exactes, hem analitzat els possibles àmbits d'identificació i les condicions que imposaven sobre els contractes.

Desenvolupament àgil. Aquest procedir, lluny de ser l'excepció ha de ser la norma: especifiquem els elements fonamentals del sistema, i comencem a dissenyar encara que quedin llacunes per especificar. Durant el disseny reanalitzem el problema allà on calgui, i completem o modifiquem les especificacions a mesura que la informació disponible ens ho permet. Aquesta és l'essència del *desenvolupament àgil*: un continu avançar amb reconsideracions a cada moment.

¹⁰ Aquí hem plantejat tot això com a axioma. Al capítol 24. ^{GLS}La idea de *participació*, pàgina 471 ho desenvolupem amb detall.

3.2 Dissenyar amb especificacions incomplertes

Desenvolupament àgil revisat. El que hem fet més amunt és completar l'especificació amb els nous elements proporcionats pel disseny, i tot seguit filtrar les propostes seguint els principis de disseny. Una alternativa hauria estat aplicar directament els principis de disseny a l'especificació incompleta. De fet aquesta és una altra manera d'aplicar el desenvolupament àgil.

Com podríem fer el disseny directament amb l'especificació incompleta de la pàgina 439?

(88) ?

Mans a l'obra. En els paràgrafs que segueixen mostrem com dissenyar directament amb especificacions incomplertes.

Identificació global. Suposem àmbit d'identificació global sobre el *Partícep*. Llavors no podem usar el *nom* com a identificador ja que el resultat no és coherent amb la semàntica donada a *Partícep*: si volem tants *partíceps* com cops una mateixa *p:Persona* apareix en el sistema, hem d'admetre l'aparició dins del sistema de *partíceps* amb el mateix nom. Per tant, en cas d'identificació d'àmbit global cal usar *identificació interna*.¹¹

Àmbits d'identificació locals possibles. Si usem el principi de la *Consistència identificadora*, l'àmbit d'identificació local de *Partícep* només pot ser la *Inscripció*.

Tipus d'identificació. Si usem la *Inscripció* com a àmbit d'identificació, el *nom* no pot ser-ne l'identificador, ja que llavors no es poden assegurar els requeriments. Per tant, en cas d'àmbit d'identificació local, no tenim més remei que usar *identificació interna*.

Del disseny a l'especificació. El disseny exigeix la *identificació interna* i proposa com a àmbit d'identificació o bé el *global* o bé el *local* a la *Inscripció* com a àmbit d'identificació. Si l'alternativa considerada semànticament no sembla massa coherent amb el concepte *Partícep*, haurem de tornar a l'especificació, cercar un concepte més adient, i modificar convenientment el MC. Ho deixem com a exercici.

4 Ús de la *Presentació*

Domini i *Presentació*. Els ES són les peticions que l'usuari fa al sistema, per sota de la capa de *Presentació*. Els seus contractes expressen les condicions d'ús

¹¹De fet aquest resultat mostra una mancança en l'anàlisi anterior. A la pàgina 429 hem bandejat l'àmbit global per violar l'exigència 3 dels requeriments. Si bé aquella afirmació era certa, només ho és en el cas de la identificació externa; però en cap moment vam considerar l'àmbit global amb identificació interna, que sempre és possible..

del *Domini* de l'aplicació. Res impedeix però que des de la *Presentació* s'imposin unes condicions més restrictives.

Objectius en el disseny del *Domini*. De cara al disseny del *Domini* cal preocupar-se de les condicions mínimes de treball tals que siguin consistents amb els requeriments, però que alhora proporcionin simplicitat i eficiència al disseny.

Objectius en el disseny de la *Presentació*. La *Presentació* ha d'usar les funcionalitats ofertes pel *Domini* per aconseguir les necessitats de l'usuari. No només s'ha de preocupar del diàleg entre l'usuari i el sistema, i d'assegurar l'acompliment de les PRE dels ES generats, sinó que també ha de "simular" els conceptes o idees que l'usuari pot tenir però que el sistema no modela directament. Aquesta simulació sovint es tradueix en un enfortiment de les condicions exigides pel *Domini*.

Exemple 95 (*Presentació* i requeriments)

Requeriments. Els requeriments exigeixen que una mateixa Persona pugui inscriure's a Caminades diferents, però que no es pugui inscriure més d'un cop a la mateixa Caminada.

Simplicitat del disseny. En el disseny, davant d'usar identificació interna i àmbit d'identificació la *Inscripció*, o d'usar la identificació externa amb àmbit d'identificació la *Caminada* o la *Inscripció*, ens hem decantat per la primera opció per ser la que compleix més principis de disseny. El resultat ha estat un disseny més simple que el que hauríem obtingut si haguéssim optat per l'altra alternativa (un sol ES enlloc de dos). La coincidència de l'acompliment dels principis de disseny amb la simplicitat del disseny resultant no és casualitat; de fet els principis de disseny cerquen la simplicitat i la flexibilitat.

El preu de la simplicitat. La identificació interna i àmbit d'identificació la *Inscripció* significa que el nom perd tota semàntica: no identifica Persona, que no apareix en el MC del domini del sistema; ni tampoc a Partícep, ja que per a Partícep usem identificació interna. D'alguna manera el disseny assumeix que l'usuari és el responsable d'assegurar la correctesa dels noms donats en el moment de crear els Partíceps. Així, la simplicitat del disseny es converteix en un conjunt d'assumpcions que la *Presentació* haurà d'assegurar.

El paper de la *Presentació*. Amb el disseny proposat la *Presentació* es pot limitar a enviar al *Domini* els ES sota les condicions dels contractes corresponents; o bé pot afegir mecanismes d'enfortiment de les condicions exigides.

Formulari d'introducció d'una nova *Inscripció*. A mode d'exemple, suposem un formulari d'introducció d'una nova *Inscripció*. Decidim que el formulari tingui una taula per a la introducció dels diversos participants; i que en confirmar la correctesa de les dades del formulari la *Presentació* genera tots els ES necessaris, en particular l'ES

22.5 Conclusions sobre el contracte

J.M. Merenciano

nouParticipant (nom, edat) per cada fila de la taula. Si afegim a la taula de participants la condició d'unicitat en el camp o columna nom, podem donar a l'usuari la sensació que el Domini treballa amb el concepte Participant i que la identificació és externa.

La Presentació i els Psartíceps. Com s'ha vist en l'exemple anterior, si usem la identificació interna dels `Partícep` llavors les restriccions exigides pels requeriments es poden introduir en *Presentació*. D'aquí que la proposta pren més força:

La proposta d'usar `Partícep`; identificació interna; i la *Inscripció* com a àmbit d'identificació de `Partícep` és una proposta vàlida

(105)(106)

5 Conclusions sobre el contracte

Propostes de l'anàlisi. De tot plegat concloem que les opcions vàlides tant per la semàntica de *Partícep* com pels principis de disseny són:¹²

- Identificació interna; àmbit d'identificació global, o local a la *Inscripció*.
 - La *Presentació* permet evitar la duplicitat de *noms* en una mateixa *Inscripció*
 - No assumeix que el *nom* identifica les *persones* (siguin aquestes o no un concepte de MC)
 - Cal assumir que l'usuari ofereix els *noms* consistents amb els requeriments¹³

Repercussions en el contracte. Pel fet d'usar *identificació interna* el contracte de `nouParticipant (nom, edat)` no ha de posar cap restricció sobre el *nom*. Per altra banda el mecanisme d'identificació interna ja assegura la unicitat del nou `Partícep`. En conclusió, la *PRE que teníem incomplerta, de fet és una PRE sobrerera!*

Contracte <code>nouParticipant (nom, edat)</code>	
Paràmetres	
• <i>nom</i> :	Nom del participant

¹²Les conclusions en el cas d'identificació global són immediates. Les exposem sense desenvolupar-les. Deixem com a exercici el seu desenvolupament.

¹³Aquesta assumpció l'avaluem a la pàgina 485.

- *edat*: Edat del participant
- PRE
1. Hi ha una *i*:Inscripció activa
- POST
1. S'ha creat una nova realització *p*:Partícep
 2. S'ha creat un enllaç entre *i* i *p*, corresponent a l'associació *inscrit* en
 3. La inscripció *i* es manté activa

6 Retorn a l'especificació

Ús d'especificacions incomplertes. En aquest capítol hem exemplificat com treballar amb una especificació incompleta o confusa. I ho hem fet des dels dos enfoc del *desenvolupament àgil*: o bé usem els principis de disseny per intentar completar l'especificació; o bé intentem dissenyar amb una especificació incompleta.

Desenvolupament en cascada. El *desenvolupament àgil* no és l'única possibilitat. El *desenvolupament en cascada* demana completar l'especificació abans d'iniciar el disseny.

Desenvolupament àgil en cascada. El *desenvolupament en cascada* tot i no tractar-se d'una opció massa recomanable per desenvolupaments sencers, dona bons resultats si es combina amb el *desenvolupament àgil*. La idea és que davant determinats problemes pot ser convenient retornar a l'especificació i explorar les diferents alternatives.

Especificació i disseny. En *desenvolupament àgil en cascada* la reconsideració de l'especificació s'ha de fer usant també els recursos del disseny. Es tracta d'una tornada enrera que vol treure valor de les diferents perspectives ofertes per l'especificació i el disseny.

Restricció del desenvolupament àgil en cascada. L'accés simultani a l'especificació i el disseny és el que obliga a usar *desenvolupament àgil en cascada* amb molta cautela i en fragments molt restringits del sistema que s'està desenvolupant.

Desenvolupament àgil en cascada de *nouParticipant* (*nom*, *edat*). Tot seguit plantejarem el disseny de *nouParticipant* (*nom*, *edat*) a través d'una re-anàlisi de l'especificació. Per fer-ho estudiarem els MC alternatius per veure quin és el que millor expressa els requeriments.

Parèntesi expositiu. Començarem però amb un petit parèntesi:¹⁴

¹⁴El capítol 23. Modelitzacions no bàsiques, pàgina 447 desenvolupa el parèntesi; el capítol 24. ^{GLS}La idea de *participació*, pàgina 471 fa l'anàlisi de l'especificació promesa.

Què significa, a nivell de disseny, tenir un concepte associatiu? I una associació n-ària?

(89) ?

Part V

Seguim explorant

Part V: Taula de continguts

23	Modelitzacions no bàsiques	447
1	Modelització d'un concepte associatiu	451
2	Conceptes febles	466
3	Associacions ternàries	469
24	^{GLS}La idea de <i>participació</i>	471
1	El concepte <i>Persona</i>	475
2	El concepte <i>Participant</i>	476
3	El concepte <i>Participació</i>	478
4	El concepte <i>Línia d'inscripció</i>	483
5	Conclusions sobre la <i>participació</i>	486
25	^{GLS}CU <i>ferInscripció</i> (2)	489
1	Punt de partida	492
2	Controlador	493
3	POST: Creació de <i>li:Línia</i>	493
4	POST: Enllaç entre <i>i i li; i activa</i>	501
5	Esdeveniment <i>fiInscripcio()</i>	501
26	Compleció semàntica	503
1	^{GLS} Un exemple: la semàntica de <i>fiInscripcio()</i>	506
2	El problema de la compleció semàntica	506
3	Principi de la <i>Incorporació tardana</i>	514
27	^{GLS}El CU <i>ferInscripció</i> amb <i>Incorporació tardana</i>	519
1	Els contractes fins al moment	522
2	Anàlisi de la <i>Incorporació tardana</i>	523
3	Un disseny amb <i>Incorporació tardana</i>	524

28	^{GLS} Cas d'ús <i>Creació de dorsals</i>	531
1	Esdeveniment <i>activaCaminada (nomC)</i>	533
2	Esdeveniment <i>creaDorsalsGrup (id)</i>	535
Historial de versions		541
	Principis	543
	Definicions	544

Capítol 23

Modelitzacions no bàsiques

1	Modelització d'un concepte associatiu	451
1.1	Un primer intent	451
1.2	Modelització de la semàntica del concepte associatiu	452
1.3	Creació d'un objecte associatiu	455
1.4	Model quasi-equivalent	459
1.5	Identificació en els conceptes associatius	463
2	Conceptes febles	466
3	Associacions ternàries	469

Contingut detallat del capítol 23

1 Modelització d'un concepte associatiu	451
1.1 Un primer intent	451
Concepte associatiu i components	451
Component i objecte associatiu	451
Semàntica d'un objecte associatiu	451
Modelització d'un objecte associatiu	451
La visibilitat associativa	451
Problemes de l'enfoc	451
1.2 Modelització de la semàntica del concepte associatiu	452
Semàntica d'un concepte associatiu	452
Context de l'anàlisi	452
Mecanisme individualitzador	452
Necessitats d'emmagatzematge	452
Consistència individualitzadora	452
Atribut que viola <i>Espill</i>	453
Doble identificació	453
El principi de la <i>Cistella</i>	453
Visibilitat indirecta	453
Repositoris i fragmentació	453
No compartició de les realitzacions de C	453
Excursió. (No compartició i composició)	453
Compartició de realitzacions de B_2	453
No necessàriament hi ha replicació	454
Dos possibles enllaços dirigits	454
1.3 Creació d'un objecte associatiu	455
Anàlisi independent	455
1.3.1 Creadors canònics assegurats	455
Candidats canònics	455
Creació canònica i emmagatzematge	455
Atribut identificador	455
Excursió. (Identificador no atributiu)	455
Fragmentació	455
Visibilitat de recuperació	455
El mètode <code>create()</code> de C	455
Visibilitats i acoblaments	455
Excursió. (Manteniment de la visibilitat invertida)	456
Unicitat de l'objecte associatiu	456
Coincidència en les conclusions	456
1.3.2 Altres creadors canònics	457
Creador canònic "extern"	457
Ús d'un creador canònic "extern"	457
Acoblaments	458
Condicions d'unicitat del concepte associatiu	458

	El perquè de la invertibilitat	458
1.4	Model quasi-equivalent	459
1.4.1	Transformació del concepte associatiu	459
	Transformació del concepte associatiu	459
	Supressió de l'associació $M - N$	459
	Obligatorietat dels extrems	459
	Quasi-equivalència	459
	Identificador del concepte C'	460
	Independència vital de C'	460
1.4.2	Modelització del model transformat	460
	Candidats canònics	460
	Creació canònica i emmagatzematge	460
	Fragmentació	460
	El mètode <code>create()</code> de C'	461
	Visibilitat de recuperació	461
	Visibilitats i acoblaments	461
	Restricció semàntica	461
	Àmbit d'identificació	461
	Visibilitats amb el creador B	461
	Conclusions en el cas d'un creador canònic assegurat	461
	Visibilitats amb el creador X	463
1.5	Identificació en els conceptes associatius	463
1.5.1	Identificació induïda	463
	Independència del creador	463
	No s'introdueixen nous acoblaments	464
	Impossibilitat de les inconsistències	464
1.5.2	Redundància en els identificadors	464
	Conext de l'anàlisi	464
	Genealogia del context	464
	Restricció semàntica	465
	Excursió. (Invariant de representació)	465
	Redundància	465
	Problemes de consistència	465
	El paper de l' <i>Estabilitat identificadora</i>	465
	Independència de realitzacions	466
	Supressió de l'atribut identificador de C	466
	El cost de la supressió de l'atribut identificador de C	466
	No paga la pena	466
2	Conceptes febles	466
	Concepte feble	466
	Àmbit d'identificació en els conceptes febles	467
	Creador d'un objecte feble	467
	Individualització genètica	467
	Fragmentació i àmbit d'identificació	468
3	Associacions ternàries	469
	Associació binària	469

Els enllaços com a tuples	469
Els enllaços dirigits com a visibilitat	469
Els enllaços ternaris	469
La visibilitat no és ordre	469
Reducció de la dimensionalitat	469
Reducció a un problema conegut	470
Diferències amb el problema conegut	470
Solució dels problemes	470
Transformació a un model equivalent	470

1 Modelització d'un concepte associatiu

Com modelitzem un concepte associatiu?

(90) ?

1.1 Un primer intent

Concepte associatiu i components. Sigui un concepte associatiu C “associat” a una associació *assoc* entre A i B . A $MComp$ tindrem tres components: A , B i C .¹

Component i objecte associatiu. Donats els tres components anteriors que modelitzen un concepte associatiu, direm que C és un *component associatiu* entre A i B . Anomenarem *objecte associatiu* a tot objecte $c : C$, realització d'un *component associatiu*.

Semàntica d'un objecte associatiu. Sigui $c : C$ un objecte associatiu, realització d'un *component associatiu* entre A i B . Això significa que existeixen dos objectes $a : A$ i $b : B$ tals que:

- modelitzen dues realitzacions $x : A$ i $y : B$
- existeix un enllaç $x-y$
- aquest enllaç és una realització de l'associació *assoc*
- el concepte associatiu C té una realització $z : C$
- $z : C$ està enllaçat a l'enllaç $x-y$

Modelització d'un objecte associatiu. Les associacions i els enllaços es modelitzen amb visibilitats d'atribut i enllaços dirigits, respectivament. Si seguim el mateix criteri, per modelitzar un objecte associatiu caldria crear un enllaç dirigit entre un objecte (el $c : C$) i un enllaç dirigit (l'existent entre $a : A$ i $b : B$).

La visibilitat associativa. Una visibilitat $A \rightarrow B$ significa que tota realització $a : A$ té la potestat per “veure” una realització $b : B$. Així, una visibilitat de C sobre la visibilitat $A \rightarrow B$ significa que tota realització $c : C$ té la potestat per veure una realització de la potestat que té tot $a : A$ de veure un $b : B$.

Problemes de l'enfoc. L'anàlisi en termes d'enllaços dirigits significaria la introducció d'un nou tipus d'enllaç dirigit: el que permetés tenir com a origen o com a destinació de l'enllaç dirigit, no pas un objecte, sinó un altre enllaç dirigit. La generalització d'aquest nou tipus d'enllaç dirigit en termes de visibilitat comporta una pèrdua clara de la intuïció que hi ha darrera el concepte de visibilitat: què vol dir que tenim coneixement d'una potestat? De què ens

¹Assumim que en el model dinàmic tenim objectes que justifiquen l'existència d'aquests tres components.

serveix? Per tot plegat concloem que la modelització dels conceptes associatius ha de seguir un altre camí.

(107)

La modelització d'un concepte associatiu en termes de visibilitat entre un component i una visibilitat, queda descartada per la seva dificultat tècnica.

1.2 Modelització de la semàntica del concepte associatiu

Semàntica d'un concepte associatiu. La semàntica d'un concepte associatiu C entre A i B és que:

- Donada una realització $z:C$ queden determinades dues realitzacions $x:A$ i $y:B$
- Donades dues realitzacions qualssevol, hi ha com a molt una única realització $z:C$ ²

Context de l'anàlisi. Anem a considerar un concepte associatiu C entre dos conceptes A_1 i A_2 . Considerem una entitat de cadascun d'aquest conceptes, i els corresponents models:

- $model(A_1) = B_1; model(A_2) = B_2$
- $model(x:A_1) = b_1 : B_1; model(y:A_2) = b_2 : B_2$
- $model(z:C) = c : C$
- $z = \langle x, y \rangle$

Mecanisme individualitzador. Donat $b_1 : B_1$, n'hi ha prou en individualitzar l'objecte $b_2 : B_2$ per individualitzar l'objecte associatiu $c : C$ que modelitza l'entitat associativa z entre les entitats x i y modelitzades per b_1 i b_2

Necessitats d'emmagatzematge. En conseqüència cal assegurar que cada $b : B_1$ tingui visibilitat de tots els C que li són pertinents; i que la selecció o individualització de cada $c : C$ es pugui fer en termes de la individualització sobre B_2 . És a dir, necessitem:

- **Visibilitat sobre el concepte associatiu.** $B_1 \rightarrow C^*$
- **Unicitat exigida pel concepte associatiu.** L'àmbit d'identificació de C és B_2 .³
- **Propietat d'identificació.** L'identificador del concepte associatiu és el de l'extrem de l'associació que no defineix l'àmbit d'identificació: $Id_C = Id_{B_2}$

²Si $x:A$ i $y:B$ estan associades segons la propietat pertinent, llavors existeix una única realització $z:C$; altrament no n'existeix cap.

³Hem usat l'extensió notacional presentada a 20.6.1.2, pàgina 414. El que estem dient és que l'àmbit d'identificació de C és el repositori propi de cada $x : B_2$.

Consistència individualitzadora. L'àmbit d'identificació proposat és coherent amb el principi de la *Consistència individualitzadora*, ja que no introdueix cap associació no present en el MC.

Atribut que viola *Espill*. El principi de la *Identificació atributiva*, juntament amb l'exigència en la propietat d'identificació acabada d'exposar, significa que el component C que modelitza el concepte associatiu C ha de tenir un atribut identificador, que a MC no apareixia.⁴

Doble identificació. @@ERRONI En el mecanisme acabat d'esbosar l'ús d'un mateix valor d'identificació ha de permetre recuperar tant l'objecte associatiu $c:C$ com l'objecte $x:B_2$ pertinent. Si bé aquesta recuperació es pot fer amb les visibilitats $B_1 \rightarrow C^*$ i $B_1 \rightarrow B_2^*$, el principi del *Baix acoblament*, així com per qüestions d'eficiència, recomanen fer-ho amb la cadena de visibilitats $A \rightarrow C^* \rightarrow B$.

El principi de la *Cistella*. @@MAL EXPLICAT Si no mantenim la visibilitat $C \rightarrow B_2$ (per exemple amb l'ús de les visibilitats $B_1 \rightarrow C^*$ i $B_1 \rightarrow B_2^*$) violem el principi de la *Cistella*. Això és així perquè el principi de la *Identificació atributiva* exigeix mantenir com a atribut de C el seu identificador, que en aquest cas també és l'identificador de B_2 . Per tant, sempre que la *cistella* contingui un objecte $c:C$ estem fent circular un identificador de B_2 ; així si $c:C$ no conté també l'objecte $x:B_2$ pertinent, estem violant el principi de la *Cistella*.

Visibilitat indirecta. De tot plegat concloem que un concepte associatiu C entre A_1 i A_2 el podem modelitzar amb una *visibilitat d'atribut indirecta* $B_1 \xrightarrow{*} B_2^*$ que té a C com a intermediari.

Repositoris i fragmentació. Amb aquesta proposta de modelització, obtenim un repositori de C fragmentat per un subconjunt de B_1 .

No compartició de les realitzacions de C . Donat un objecte associatiu $c:C$ aquest "defineix" unívocament un objecte $x:B_1$ i un objecte $y:B_2$. Això significa que un mateix objecte $c:C$ no es pot compartir en els repositoris mantinguts per diferents B_1 .

Excursió. (No compartició i composició) Si no hi ha d'altres rèpliques del repositori de C llavors podem dir que cada $x:B_1$ és *propietari* de les "seves" C , que no són visibles per ningú més. Així, i sempre que els principis de creació ho refermin, podríem expressar la visibilitat $B_1 \rightarrow C^*$ com una *composició*.

⁴Les realitzacions dels conceptes associatius s'identifiquen a partir de la identificació d'un enllaç; no tenen identificació pròpia.

Compartició de realitzacions de B_2 . Les realitzacions de B_2 són anteriors a les de C : si usem la notació que hem usat en exposar la semàntica d'un objecte associatiu⁵ tenim que un $z:C$ només pot existir si existeix l'enllaç dirigit $x:B_1 \rightarrow y:B_2$, la qual cosa exigeix l'existència tant de $x:b_1$ com de $y:B_2$. Per tant si la visibilitat $C \rightarrow B_2$ de recuperació de B_2 la veiem com una visibilitat d'emmagatzematge, necessàriament la $y:B_2$ pertinent l'hem d'haver obtinguda d'un altre repositori; es tracta per tant d'una realització compartida. És a dir les realitzacions de B_2 són *compartides*.

No necessàriament hi ha replicació. És possible que hi hagi alguna B_2 que no participi amb cap B_1 en la construcció d'una C . Per aquest motiu no podem afirmar que el repositori de B_2 és replicat. Si l'associació pertinent entre A_1 i A_2 és *obligada* per l'extrem de la A_1 llavors sí que el repositori de B està *replicat*.⁶

Dos possibles enllaços dirigits. El mecanisme de modelització exposat és simètric: enlloc d'usar B_1 com a àmbit d'identificació del component associatiu, podem usar B_2 . Per tant tant podem modelitzar amb la visibilitat $B_1 \xrightarrow{*} B_2^*$ que té a C com a intermediari, com amb la visibilitat $B_2 \xrightarrow{*} B_2^*$ que té a C com a intermediari (o fins i tot amb ambdues visibilitats simultàniament). És responsabilitat del disseny decidir quina és la visibilitat més adient.

(108)

Modelització d'un concepte associatiu (1)

Per modelitzar un concepte associatiu C entre A i B usem:

- *Visibilitat d'atribut indirecta i multiavaluada* de A a B (o viceversa), que passa per C
- *L'àmbit d'identificació* de C és el de l'origen de la visibilitat indirecta emprada per a la modelització
- La *propietat d'identificació* sobre aquest àmbit d'identificació és la *propietat d'identificació* de la destinació de la visibilitat indirecta emprada per a la modelització
- Cal un *atribut identificador* a C

⁵Vegeu l'apartat 23.1.1., pàgina 451.

⁶Aquesta darrera afirmació no és certa del tot: es basa en l'exigència de l'existència prèvia de $y:B_2$ en el moment de crear $z:C$. L'obligatorietat en l'associació, però, introdueix una circularitat de dependències: la $y:B_2$ ara no pot existir sense la $z:C$! Per resoldre el problema cal usar el mecanisme de les *transaccions* que permet que de manera temporal no es compleixi alguna de les exigències. D'aquesta manera es poden construir $y:B_2$ i $z:C$ de manera simultània (és a dir, dins d'una mateixa transacció).

El resultat és que ha desaparegut el requeriment de l'existència prèvia de la $b:B$, i per tant no és necessària la rèplica del repositori de B . En el text el que volem remarcar és que en el cas que tota $b:B$ tingui una $c:C$ associada, llavors cada $b:B$ apareix dins d'una $c:C$ o dins del repositori de totes les B . El que diem en aquesta nota és que els mecanismes d'implementació poden fer innecessari un repositori amb totes les B , si aquestes ja apareixen dins de les $c:C$.

1.3 Creació d'un objecte associatiu

Anàlisi independent. A partir de l'anàlisi de la semàntica d'un objecte associatiu, i jugant amb els àmbits d'identificació, hem obtingut una proposta per a la modelització dels conceptes associatius. Tot seguit realitzarem una anàlisi independent basada principalment en els principis del disseny.

Qui és el responsable de crear un objecte associatiu?

(91) ?

1.3.1 Creadors canònics assegurats

Candidats canònics. Tot concepte associatiu C involucra una associació entre dos conceptes A i B . Per tant, per a la creació d'un objecte $c : C$ sempre tindrem dos candidats canònics: A i B .

Creació canònica i emmagatzematge. Suposem que usem A com a creador canònic de C . Per tant A és *emmagatzemador* de C , que es manté mitjançant una *visibilitat d'emmagatzematge* $A \rightarrow C^*$.

Atribut identificador. La multiplicitat de la visibilitat d'emmagatzematge exigeix un mecanisme per individualitzar cadascun dels objectes $c : C$ emmagatzemats. És a dir, cal afegir un *atribut identificador* a C .

Excursió. (Identificador no atributiu) En certs casos la individualització d'un col·lectiu es pot fer mitjançant identificadors no atributius. Per exemple, si el multiojecte responsable de l'emmagatzematge s'implementa amb una llista es pot usar la posició dins la llista per individualitzar els diferents objectes. Aquest mecanisme però no és gens robust i si de cas només aporta, i no pas sempre, alguna millora en eficiència.

El principi de la *Identificació atributiva* intenta evitar justament aquest tipus d'eficiències prematures.

Fragmentació. La visibilitat d'emmagatzematge $A \rightarrow C^*$ significa que el repositori de C està fragmentat; cada $a : A$ n'és un fragment.

Visibilitat de recuperació. En cada fragment del repositori de C s'emmagatzemen les C corresponents a diferents B . Això exigeix mantenir una *visibilitat de recuperació* de les B : $C \rightarrow B$.

El mètode `create()` de C . Per tal de mantenir la visibilitat $C \rightarrow B$ cal que el creador $a : A$ envii el $b : B$ com a argument del missatge `create()` de C .

Visibilitats i acoblaments. En conclusió tenim una visibilitat $A \rightarrow C^* \rightarrow B$. A més A ha de tenir algun tipus d'acoblament amb B .

Excursió. (Manteniment de la visibilitat invertida) En rebre el missatge `create(b)`, el component `C'` pot delegar en el paràmetre la tasca de mantenir l'enllaç:

```
class C                                class B
{
    C(B:b)
    {
        b.assigna(this);
    }
}

                                {
    List<C> mc;
    void assigna(C:x)
    {
        mc.add(x);
    }
}
```

El resultat és una visibilitat $B \rightarrow C^*$, i una dependència de paràmetre $C \xrightarrow{p} B$.

Amb aquest mecanisme l'acoblament és superior i es perd la capacitat de recuperar el `b:B` pertinent a partir de cada `c:C`. Per tant no sembla una bona solució. Malgrat tot en determinades circumstàncies pot ser una alternativa a considerar.

Unicitat de l'objecte associatiu. Fins aquí no hem fet intervenir les condicions de la semàntica d'un concepte associatiu.⁷ En fer-les intervenir cal exigir que en el repositori de cada `a:A` no hi hagi repetit un mateix `b:B`. És a dir, cal exigir que l'àmbit d'identificació de `C` sigui `A`; i que l'identificador sigui el mateix que `B`.

Coincidència en les conclusions. L'intent de plasmar la semàntica del concepte associatiu en termes de `MComp` ens ha portat a l'exigència d'unes determinades visibilitats, fragmentacions i àmbits d'identificació. L'anàlisi del mecanisme de creació d'un objecte associatiu ens ha portat exactament a les mateixes conclusions.

(109)

Modelització d'un concepte associatiu (2)

La modelització d'un concepte associatiu tal com s'ha expressat a la conclusió 108, pàgina 454, queda refermada per l'aplicació dels principis de disseny.

A més sabem que:

- Donada la visibilitat indirecta emprada per la modelització, el seu extrem que defineix l'àmbit d'identificació té algun tipus d'acoblament amb l'altre extrem d'aquesta visibilitat
- Donada la visibilitat indirecta emprada per la modelització, el concepte associatiu té dependència de paràmetre sobre l'extrem d'aquesta visibilitat que defineix la propietat d'identificació

⁷Vegeu l'apartat 23.1.2., pàgina 452.

Exemple 96 (Modelització d'un concepte associatiu) *En un Albarà de compra hi consten múltiples Articles. La quantitat de cada Article que es ven en un mateix Albarà és un atribut del concepte associatiu Línia entre Albarà i Article.*

Cal observar com aquest model no permet que un mateix article aparegui en línies diferents dins d'un mateix albarà.

Per a la modelització usarem els components Albarà, Article i Línia.

Com a creador canònic de la Línia usarem l'Albarà. Per tant tenim la visibilitat d'emmagatzematge $\text{Albarà} \rightarrow \text{Línia}^$, i la visibilitat de recuperació $\text{Línia} \text{Article}$.*

L'àmbit d'identificació de la Línia és l'Albarà, i ha de coincidir amb l'identificador d'Article.

Tot plegat permet escriure en Java la següent classe:

```
class Linia
{
    IdArticle id; // El tipus de l'identificador
                // de línia és el d'un
                // identificador d'article
    Article a;

    int quant; // Quantitat d'articles venuts

    Linia(Article a, int quant)
    {
        id = a.id;
        this.a = a;
        this.quant = quant;
    }
}
```

1.3.2 Altres creadors canònics

Creador canònic “extern”. En l'anàlisi prèvia hem considerat els candidats canònics que necessàriament sempre existeixen. Però és possible que existeixin d'altres candidats: n'hi ha prou en què existeixi una associació entre un concepte X i un concepte associatiu C . En el que segueix direm que X és un *candidat canònic “extern”* de C , en el sentit que no és un candidat canònic definit per C com a concepte associatiu.

Ús d'un creador canònic “extern”. Si usem X com a creador canònic de C , també n'és *emmagatzemador*, i això exigeix una *visibilitat d'emmagatzematge* $X \rightarrow C^*$, i un *atribut identificador* per a C . Per tant tenim un repositori de C fragmentat per X . En cada fragment, però, hi conviuen $c : C$ corresponents a dife-

rents $a:A$ i a diferents $b:B$. En conseqüència cal mantenir dues *visibilitats de recuperació*: $C \rightarrow A$ i $C \rightarrow B$.

Acoblaments. Per tal de mantenir les *visibilitats de recuperació* cal que el creador $d:X$ tingui visibilitat dels objectes $a:A$ i $b:B$ corresponents, i els passi com arguments del missatge `create()` de C . Per tant l'acoblament de X és gran: visibilitat d'atribut sobre C i algun tipus de visibilitat sobre A i B . Per contra, tant A com B tenen un acoblament mínim.

Condicions d'unicitat del concepte associatiu. Les condicions de la semàntica sobre un component associatiu⁸ en aquest cas exigeixen que l'àmbit d'identificació sigui la C , i que l'identificador sigui una *funció invertible* sobre els identificadors de A i B :

$$Id_C = f(Id_A, Id_B)$$

El perquè de la invertibilitat. La funció que defineix l'identificador de C en funció dels identificadors de A i B ha de ser invertible per assegurar que un mateix identificador de C es correspongui a un sol parell $\langle id_A : Id_A, id_B : Id_B \rangle$; pel fet de ser una funció assegurem que donat un parell $\langle id_A : Id_A, id_B : Id_B \rangle$ només hi ha un valor $id_C : Id_C$ possible. Aquestes dues condicions són precisament les que ens demana la semàntica d'un concepte associatiu.

Exemple 97 (Identificació per concatenació)

Sigui una Empresa que té diferents Departaments i que contracta diferents Empleats. Cada Empleat s'assigna a un Departament. El sou depèn de l'Assignació; és a dir, és un atribut concepte associatiu Assignació entre Empleat i Departament.

Suposem que decidim usar Empresa com a creador d'Assignació. És a dir, mantenim la visibilitat $Empresa \rightarrow Assignació^*$. Per tal d'identificar les Assignacions ens cal definir una funció invertible que prengui com a arguments els identificadors de l' $e:Empleat$ i del $d:Departament$ pertinents.

Un mecanisme molt habitual per construir aquesta funció d'identificació és la concatenació marcada. En aquest cas, i considerant que, en virtut de la Identificació atributiva, tot component té un atribut textual id amb el seu identificador; que "+" és la concatenació de cadenes de text; i que les cometes (") expressen un literal textual, podem definir:

$$c.id = e.id + "#" + d.id$$

⁸Vegeu l'apartat 23.1.2., pàgina 452.

Modelització d'un concepte associatiu (3)

Un concepte associatiu es pot modelitzar com s'indica a la conclusió 109, pàgina 456. Alternativament per modelitzar un concepte associatiu C entre A i B podem:

- *Visibilitat d'atribut indirecta* des d'algun altre component X sobre A , amb C com a intermediari: $X \rightarrow C^* \rightarrow A$
- *Visibilitat d'atribut indirecta* des d'algun altre component X sobre B , amb C com a intermediari: $X \rightarrow C^* \rightarrow B$
- L'àmbit d'identificació de C és X
- La propietat d'identificació de C és una funció invertible que pren com a arguments els valors dels identificadors dels objectes $a:A$ i $b:B$ pertinents: $Id_C = f(Id_A, Id_B)$
- Cal un atribut identificador a C

1.4 Model quasi-equivalent

1.4.1 Transformació del concepte associatiu

Transformació del concepte associatiu. Tot MC que contingui un concepte associatiu es pot transformar en un MC quasi-equivalent que no contingui cap concepte associatiu.

Supressió de l'associació $M - N$. Sigui un concepte associatiu C entre A i B . Això significa que entre A i B hi ha una associació *assoc*, $M - N$,⁹ i que C és una "associació" sobre *assoc*. El model transformat converteix C en un concepte C' , amb una associació $1 - N$ entre A i C' ; i una altra associació $N - 1$ entre C' i B . En el model transformat *assoc* desapareix.¹⁰

Obligatorietat dels extrems. En el model de partida tota realització del concepte C només pot existir en relació a una realització de A i una realització de B . Per aquest motiu en el model transformat l'associació entre A i C' és obligada per l'extrem de l' A ; i l'associació entre B i C' és obligada per l'extrem de la B .

Quasi-equivalència. El model de partida i el transformat no són equivalents: per ser-ho cal introduir restriccions semàntiques en el model transformat, en concret que no hi pot haver dues realitzacions del concepte C' diferents que

⁹Els conceptes associatius sobre associacions $1 - N$ són possibles, però molt poc habituals: en la majoria dels casos són el símptoma d'un mal disseny. En el cas que l'associació sigui realment $1 - N$, tot el que plantejem aquí continua sent vàlid: n'hi ha prou en suposar l'associació és un cas particular d'una associació $M - N$.

¹⁰De fet *assoc* i C són el mateix en el model de partida: *assoc* és la visió del concepte associatiu com a associació, C n'és la visió com a concepte; C són els atributs aplicats a l'associació *assoc*.

s'associïn al mateix parell de realitzacions de A i B . És a dir, cal introduir com a restricció semàntica la unicitat exigida per la semàntica del concepte associatiu.¹¹

Identificador del concepte C' . En el model transformat C' és un concepte, i per tant ha de tenir identificador propi.¹² En el model de partida el concepte associatiu C s'identifica a partir de la identificació dels extrems de l'associació *assoc*. Així ambdós models no només no comparteixen la mateixa semàntica (són *quasi-equivalents*) sinó que a més en el model transformat hi apareix un atribut més: l'identificador de C' .

Independència vital de C' . Pel fet de tractar-se d'un concepte amb dret propi, C' pot existir de manera independent a A i B . En canvi tota realització del concepte associatiu C només pot existir en relació a una realització de A i una realització de B . El concepte associatiu C és *dependent*; el transformat C' és *independent*. És l'*obligatorietat* en els extrems A i B de les dues associacions en les que intervé C' qui reintrodueix la dependència de C' respecte A i B .

(111)

Transformació de models

La transformació d'un MC amb conceptes associatius a un model MC' sense conceptes associatius proporciona un model equivalent només si s'hi afegeixen les restriccions semàntiques exigides per la semàntica dels conceptes associatius, expressades a la pàgina 452.

Exemple 98 (Transformació d'un concepte associatiu) *Sigui l'exemple 96, pàgina 457. La transformació del model, via la supressió del concepte associatiu, fa aparèixer un concepte Línia amb dos atributs: l'identificador i la quantitat.*

La Línia té una associació $N - 1$ amb Article, i una altra associació $N - 1$ amb Albarà.

1.4.2 Modelització del model transformat

Candidats canònics. En el model transformat els candidats a ser creadors canònics de C' són A , B , i eventualment tot X tal que hi hagi una associació entre X i C' .

Creació canònica i emmagatzematge. L'ús de A com a creador canònic de C' exigeix una visibilitat d'emmagatzematge $A \rightarrow C ; *$.

¹¹Vegeu l'apartat 23.1.2., pàgina 452.

¹²MC no posa cap restricció ni exigència sobre aquest identificador. Una altra cosa és que per altres motius, principalment d'implementació, el desenvolupador decideixi construir l'identificador del component C com la concatenació dels identificadors de A i B . D'aquesta manera compleix la necessitat d'introduir un identificador alhora que hi introdueix la restricció semàntica exigida per la semàntica del concepte associatiu.

Fragmentació. La visibilitat d'emmagatzematge $A \rightarrow C' *$ significa que el repositori de C' està fragmentat; cada $a:A$ n'és un fragment.

El mètode `create()` de C' . L'obligatorietat en els extrems de les associacions exigeix que el creador $a:A$ envii el $b:B$ com a argument del missatge `create()` de C' .

Visibilitat de recuperació. L'ús d'un argument $b:B$ en el mètode `create()` de C' porta a la necessitat d'una visibilitat de recuperació $C' \rightarrow B$.

Visibilitats i acoblaments. En conclusió tenim una visibilitat $A \rightarrow C' * \rightarrow B$. A més A ha de tenir algun tipus d'acoblament amb B .

Restricció semàntica. La restricció semàntica imposada per la semàntica del concepte associatiu es pot introduir en els contractes dels ES,¹³ i d'aquesta manera traspasar la responsabilitat del seu manteniment a l'usuari o client; o bé es pot introduir a MComp en termes de restricció dels àmbits d'identificació.

Àmbit d'identificació. En ser C' independent de A i B , en principi no podem restringir l'àmbit d'identificació de C' ni a A ni a B . La proposta de fragmentar el repositori de C' per A s permet pensar en definir A com l'àmbit d'identificació de C' . I llavors es pot introduir la restricció semàntica imposada per la semàntica del concepte associatiu a base d'exigir que l'identificador de C' sigui justament l'identificador de B . El resultat és que C' perd la seva independència: com a mínim l'identificador de cada $c:C$ ha de ser el d'un $b:B$.

Visibilitats amb el creador B . L'ús de B com a creador canònic és el cas simètric al del creador A . Cal una visibilitat $B \rightarrow C' * \rightarrow A$, i algun tipus d'acoblament entre A i B .

Conclusions en el cas d'un creador canònic assegurat. En conclusió, l'ús com a creador de C' d'un dels candidats canònics assegurats (A o B) porta al mateix MComp que l'obtingut en la modelització directa del concepte associatiu.¹⁴

Modelització del model transformat sense conceptes associatius

Donat un model MC' que s'ha obtingut com la transformació, via supressió del concepte associatiu C entre A i B , d'un model MC , i tal que $model(MC) = MComp$ i $model(MC') = MComp'$ tenim que:

- A i B apareixen en ambdós models $MComp$ i $MComp'$

(112)

¹³Amb això no n'hi ha prou. De fet cal introduir la restricció en els invariants dels components afectats, la qual cosa obliga a introduir-la en els contractes de totes les seves operacions.

¹⁴Cal observar com l'atribut identificador que la transformació del model exigeix a C' és el mateix atribut que en la modelització directa del concepte associatiu cal per individualitzar de la visibilitat d'emmagatzematge.

- On a MComp apareix C a MComp' apareix C' , idèntic a C però amb un atribut més que fa d'identificador
- Les visibilitats, acoblaments i fragmentacions són les mateixes en ambdós models MComp i MComp'
- L'àmbit d'identificació de C a MComp és A (o B , segons el creador emprat). L'àmbit d'identificació de C' a MComp' no necessàriament ha d'estar restringit a A o a B

La introducció de la restricció semàntica exigida per l'equivalència de MC i MC'^a obliga a exigir que l'àmbit d'identificació de C' a MComp' sigui A (o B , segons el creador emprat), i que la propietat d'identificació de C' sigui la mateixa que la de B (o A , respectivament). El resultat és que MComp i MComp' són idèntics.

^aVegeu la conclusió 111, pàgina 460.

Exemple 99 (Modelització del model transformat) *Sigui el model transformat obtingut en l'exemple 98, pàgina 460.*

Aquest model, permet que un mateix article aparegui en línies diferents dins d'un mateix albarà, en contra del que diu el model original de l'exemple 96, pàgina 457.

Per a la modelització usarem els components Albarà, Article i Línia.

Com a creador canònic de la Línia usarem l'Albarà. Per tant tenim la visibilitat d'emmagatzematge $\text{Albarà} \rightarrow \text{Línia}^$, i la visibilitat de recuperació LíniaArticle .*

L'àmbit d'identificació de la Línia moment no el tenim restringit. Usarem un mecanisme d'identificació interna sense especificar-ne el funcionament,

*Tot plegat permet escriure en Java la següent classe (on hem marcat amb * els canvis respecte els de l'exemple 96):*

```
class Linia
{
    IdLinia    id;                // *
    Article    a;
    int        quant;            // Quantitat d'articles venuts

    Linia(Article a, int quant)
    {
        id        = creaID();    // *
        this.a     = a;
        this.quant = quant;
    }
}
```

```
}  
}
```

Exemple 100 (Introducció de la restricció semàntica) *Sigui l'exemple 99. La introducció de la restricció semàntica exigida per la semàntica de la `Línia` com a concepte associatiu, força a què l'identificador de la `Línia` sigui el de l'`Article` sobre el que té visibilitat. El resultat és un model idèntic al de l'exemple 96, pàgina 457.*

És important observar com les exigències sobre l'àmbit d'identificació i sobre la propietat d'identificació fan que ara ja no es pugui repetir un mateix article en línies d'un mateix albarà.

Visibilitats amb el creador `X`. L'ús de `X` com a creador canònic de `C` exigeix una visibilitat d'emmagatzematge $X \rightarrow C^*$. L'obligatorietat de les associacions exigeix que el missatge `create()` de `C` rebi com a arguments el $a:A$ i $b:B$ pertinents; la qual cosa porta a la necessitat de les visibilitats de recuperació $C \rightarrow A$ i $C \rightarrow B$. Finalment, la introducció de la restricció semàntica exigida per la semàntica del concepte associatiu porta a definir l'identificador de `C` en termes de l'identificador de `A` i `B`. El resultat és un model idèntic a l'eposat a la conclusió 110, pàgina 459.

De la quasi-equivalència a l'equivalència

La supressió dels conceptes associatius d'un MC, via transformació del model, porta a un model quasi-equivalent.

La introducció de les restriccions necessàries per obtenir l'equivalència dels models conceptuais, assegura que els models de components també són equivalents.

(113)

1.5 Identificació en els conceptes associatius

1.5.1 Identificació induïda

Independència del creador. Independentment de quin creador canònic s'usi per a crear un concepte associatiu, la semàntica del propi concepte associatiu exigeix *identificació induïda*: l'identificador ha de ser l'identificador de l'objecte rebut com a argument del missatge `create()`.¹⁵

¹⁵En el cas d'usar un creador "extern", l'identificador del concepte associatiu s'obté com a el valor d'una funció invertible aplicada a l'identificador dels objectes rebuts com a argument del missatge `create()`.

(114)

La modelització d'un *concepte associatiu* exigeix *identificació induïda*

No s'introdueixen nous acoblaments. En aquest cas particular, però, la *identificació induïda* no és problemàtica, ja que no introdueix nous acoblaments. Els arguments del `create()` de *C* són necessaris per mantenir les interrelacions que *c:C*, com a concepte associatiu, ha de tenir amb *a:A* i amb *b:B*.

Impossibilitat de les inconsistències. En l'anàlisi de la *identificació induïda*¹⁶ s'ha vist com aquesta pot generar inconsistències. Aquí aquest no és el cas. L'identificador del concepte associatiu és el d'un altre objecte ja creat; per tant si en ell no es donen inconsistències en la identificació, i ningú no viola els contractes de les operacions de creació del concepte associatiu, la inconsistència no és possible.

Exemple 101 (Consistència en la identificació induïda) *Sigui l'exemple 96, pàgina 457. La creació d'una nova Línia exigeix enviar el missatge de creació amb un Article com a argument (a part de l'atribut quantitat de la Línia):*

```
Article a = obtenirArticle();  
....  
Linia l = new Linia (a, 3);    // Es venen 3 articles "a"
```

Per una banda cal suposar que $a:Article$ no és inconsistent: l'hem obtingut correctament del sistema. Per l'altra l'operació que respon al missatge de creació d'una Línia ha de tenir una PRE que asseguri la consistència de l'àmbit d'identificació:

PRE: No existeix en l'àmbit d'identificació cap Línia amb el mateix identificador

Per tant, si ningú viola el contracte em missatge `create(a, 3)` enviat a Línia (a través del `new` en el cas del Java), no es pot produir cap inconsistència.

1.5.2 Redundància en els identificadors

Condeix de l'anàlisi. Sigui un concepte associatiu *C* entre *A* i *B*. Considerem la modelització amb un component *C* amb un atribut `id`, una visibilitat d'emmagatzema una visibilitat d'emmagatzematge $A \rightarrow C^*$, i una visibilitat de recuperació $C \rightarrow A$.

Genealogia del context. L'anàlisi que segeueix tindrà en compte el context acabat d'exposar. En cas d'usar *B* com a creador canònic de *C*, l'anàlisi és

¹⁶Vegeu el capítol 15. Identificadors, pàgina 277.

idèntica. En cas d'usar un component X com a creador canònic de C , l'anàlisi és molt similar.

Restricció semàntica. Per tal que el model sigui correcte cal exigir que l'identificador de cada $c:C$ sigui el de l'objecte $b:B$ amb el que està enllaçat via la visibilitat de recuperació $C \rightarrow B$.

Excursió. (Invariant de representació) En termes d'implemenació la restricció semàntica sobre la igualtat dels identificadors, pot aparèixer com a *invariant de la representació*.

Per exemple, sugui el cas de la *línia de compra* com a concepte associatiu (exemple 96, pàgina 457). Introduïm l'invariant de la representació en forma de comentari (línia marcada amb *).

```
class Linia
{
    // INV:  a.id == id          // *

    IdArticle  id;  // El tipus de l'identificador
                  // de línia és el d'un
                  // identificador d'article
    Article    a;

    int        quant;  // Quantitat d'articles venuts

    Linia(Article a, int quant)
    {
        id      = a.id;
        this.a   = a;
        this.quant = quant;
    }
}
```

Redundància. La necessitat d'introduir la restricció semàntica sobre la igualtat dels identificadors ésconseqüència immediata de la *redundància* del model: el mateix valor identificador apareix com a atribut de $c:C$ i com a atribut de $b:B$.

Problemes de consistència. La redundància en la informació té un problema afegit: el de mantenir la *consistència* dels valors redundants. Un canvi en un atribut ha de comportar el canvi en tots els atributs que mantenen la mateixa informació.

El paper de l'Estabilitat identificadora. El principi de l'*Estabilitat identificadora* diu que donat el valor a un atribut identificador, aquest valor ja no pot variar. En el cas que ens ocupa, com que la redundància es dona en atributs identificadors, i l'*Estabilitat identificadora* ens assegura que els seus valors no

canvien, no cal que ens preocupem dels problemes de consistència causats per la redundància.

? (92)

És possible suprimir la redundància dels identificadors en el model d'un concepte associatiu?

Independència de realitzacions. Les realitzacions de *B* són independents de les de *C*; de fet són prèvia a aquestes. A més, en general, poden existir realitzacions $b : B$ que no estiguin associades a cap $c : C$. Per tant no podem suprimir l'atribut identificador de *B*.

Exemple 102 (Independència de realitzacions) Tornem a l'exemple de la Línia de compra com a concepte associatiu (exemple 96, pàgina 457).

Els Articles existeixen abans de ser comprats. Hi pot haver Articles que no compra ningú, i que per tant no en tenim visibilitat des de cap Línia.

Per tant és evident que cal una manera d'individualitzar els diferents Articles sense necessitat de passar per cap Línia.

Supressió de l'atribut identificador de C. Podem suprimir l'atribut identificador de *C*? Si sabem quina és la visibilitat d'atribut que defineix l'identificador¹⁷ en principi la supressió de l'atribut identificador de *C* és possible.

El cost de la supressió de l'atribut identificador de C. El preu que cal pagar per la supressió de l'atribut identificador de *C* és en claredat i en eficiència. Perdem en *claredat* ja que l'atribut identificador de *C* no apareix de forma explícita, i caldrà afegir algun comentari per expressar d'onj s'ha d'obtenir. Perdem en *eficiència* ja que tota selecció sobre un multiobjecte de *C* requereix una indirecció cap a *B* per accedir a l'identificador.

No paga la pena. El principi de l'*Eficiència en segon pla* ens diu que en un primer estadi no ens hem de preocupar de l'eficiència. Per altra banda, un dels objectius del disseny és obtenir un model de la solució que sigui simple, clar i mantenible. En aquest sentit preferim pagar el cost de la redundància identificadora, que el cost de la seva supressió.

2 Conceptes febles

Concepte feble. Un *concepte feble* *B* és aquell que només té sentit en relació a un altre concepte *A*, que anomenem *fort*.¹⁸ La identificació de la realització

¹⁷El model exigeix la visibilitat $C \rightarrow B$, però res no impedeix l'existència d'una altra visibilitat CY .

¹⁸Els conceptes febles són molt habituals en els models Chen. En canvi, en UML acostumen a quedar diluïts en d'altres estructures, com la composició.

d'un concepte feble significa identificar la realització corresponent del concepte fort pertinent, més la identificació dins del context del concepte fort. De fet el valor d'identificació dins d'un context pot coincidir amb el d'altres valors d'identificació, sempre que facin referència a contextos diferents.

Exemple 103 (Concepte feble) *A l'exemple 96, pàgina 457, s'ha presentat la Línia de compra com un concepte associatiu entre l'Albarà i l'Article.*

A l'exemple 98, pàgina 460, s'ha presentat la Línia de compra com un concepte independent associat a l'Albarà i a l'Article.

En aquest segon model conceptual, en un mateix Albarà es pot comprar el mateix Article en Línies diferents, cosa que el primer model conceptual prohibeix.

Un tercer model conceptual possible és el que defineix la Línia com un concepte feble respecte de l'Albarà. Això significa que l'identificador de la Línia només té sentit dins l'Albarà. Per exemple, podem usar com a identificador de Línia un número d'ordre dins de l'Albarà: el númeroDeLínia.

L'identificador propi d'una Línia (el seu número) pot coincidir amb el d'altres Línies, però dins del context d'un a:Albarà concret, no hi pot haver dues Línies amb el mateix número.

L'ús de la Línia com a concepte feble permet que en Línies diferents es pugui comprar el mateix Article. En aquest sentit el model es comporta de manera idèntica al segon dels models exposats més amunt (el de la Línia com a concepte independent).

Àmbit d'identificació en els conceptes febles. Un concepte feble imposa ja a MC que l'àmbit d'identificació del concepte feble sigui el concepte fort pertinent.

Creador d'un objecte feble. Sigui un concepte feble C sobre un concepte fort A. El responsable de crear l'objecte $c:C$ és necessàriament A: és creador canònic (tot i que potser no l'únic), i és expert en la individualització.

Exemple 104 Creador d'un objecte feble *Sigui l'exemple 103. Tant l'Albarà com l'Article són candidats a creador canònic. Però només l'Albarà té tota la informació d'expertesa, en concret, el és ell qui coneix l'àmbit d'identificació: el repositori de les seves Línies.^a Per tant el creador de la Línia és l'Albarà.*

^aEn la generació per comprovació l'àmbit d'identificació s'usa com a repositori de contrast.

Individualització genètica. La creació dels objectes febles és consistent amb el principi de la *Individualització efectiva*: el concepte fort pertinent, és el creador i l'individualitzador del concepte feble; estem, per tant, en *individualització*

genètica.¹⁹

Fragmentació i àmbit d'identificació. Donat un concepte feble *C* sobre un concepte fort *A*, per definició de concepte feble, necessàriament l'àmbit d'identificació de *C* és *A*. Per tant a MComp tenim un repositori de *C* fragmentat per *A*.

(115)

Modelització de conceptes febles

Per modelitzar un concepte *B* feble respecte *A* usarem:

- Dos components *A* i *B*
- L'àmbit d'identificació de *B* és *A*
- El creador canònic de *B* és *A*

Exemple 105 Comparació del models Considerem els tres possibles models conceptuals per a la Línia de compra analitzats fins ara:

Model associatiu La Línia és un concepte associatiu entre l'Albarà i l'Article. (Exemple 96, pàgina 457.)

Model independent La Línia és un concepte independent, associat amb l'Albarà i l'Article. (Exemple 99, pàgina 462.)

Model feble La Línia és un concepte feble respecte amb l'Albarà, i que està associada amb l'Article. (Exemple 103, pàgina 467.)

Semànticament el model independent i el feble són equivalents: ambdós permeten comprar un mateix Article en diferents Línies d'un mateix Albarà. Des d'un punt de vista del procés d'identificació, però, hi ha algunes diferències subtils. El següent quadre mostra la comparativa dels tres models:

Model	Creador de Línia	Àmbit d'identificació de Línia	Tipus d'identificació
Associatiu	Albarà o Article	El creador	Induïda
Independent	Albarà o Article	Independent	Genètica
Feble	Albarà	El creador	Genètica

El model feble per està entremig dels altres dos models. Permet la repetició Articles en una mateixa compra; però tant la creació com l'àmbit d'identificació de Línia està lligada a l'Albarà; no hi ha independència.

¹⁹De fet, tenim les condicions per a fer la individualització genètica. Però res no ens obliga (a excepció del principi de la Individualització efectiva) a fer ús d'aquestes condicions.

L'identificador propi de $Línia$ pot ser substituït, en la implementació, per les propietats d'ordre de l'estructura d'emmagatzematge. Per exemple, la visibilitat d'atribut $Albarà \rightarrow Línia^$ es pot implementar amb un atribut d' $Albarà$ de tipus $List<Línia>$; la posició de cada $l:Línia$ dins la llista és el seu valor d'identificador dins del context de l' $a:Albarà$ corresponent.*

3 Associacions ternàries

Com modelitzem una associació ternària $M - N - P$ entre A, B i C ?

(93) ?

Associació binària. La modelització d'una associació binària $M - N$ entre A i B és una visibilitat d'atribut $A \rightarrow B^*$ (o la inversa). És a dir, tot enllaç $a:A - b:B$ el convertim en un enllaç dirigit cap a un dels dos extrems de l'enllaç.

Els enllaços com a tuples. Un enllaç binari $a:A-b:B$ el podem veure com un 2-tuple $\langle a:A, b:B \rangle$. La bidimensionalitat del 2-tuple permet dibuixar en el pla els enllaços binaris.

Els enllaços dirigits com a visibilitat. L'enllaç dirigit imposa una visibilitat: és des de $a:A$, per exemple, que accedim a $b:B$; donat $a:A$, $b:B$ queda determinat. La bidimensionalitat dels enllaços binaris permet expressar aquesta visibilitat en termes d'orientació d'un segment del pla.

Els enllaços ternaris. Un enllaç ternari es pot veure com un 3-tuple $\langle a:A, b:B, c:C \rangle$. La tridimensionalitat del 3-tuple impedeix la seva representació en el pla.

Què significa la visibilitat en un enllaç dirigit ternari?

(94) ?

La visibilitat no és ordre. L'ordre en el 3-tuple no soluciona el problema. Dir que el coneixement de $a:A$ permet accedir a $b:B$ i que aquest permet accedir a $c:C$ potser pot servir per modelitzar l'enllaç $\langle a:A, b:B, c:C \rangle$; però la multiplicitat de l'associació permet l'existència d'una altre enllaç $\langle x:A, b:B, y:C \rangle$. En aquest cas, un cop hem accedit a $b:B$ com sabem si l'enllaç dirigit "segeuix" cap a $c:C$ o cap a $y:C$?

Reducció de la dimensionalitat. La idea per a modelitzar un enllaç ternari (i en general un n-ari) és la reducció de la dimensionalitat de l'enllaç. Sigui l'enllaç $\langle a:A, b:B, c:C \rangle$; donat el coneixement de $a:A$ quin és el coneixement que ens manca per obtenir el model de l'enllaç? Òbviament el que ens manca és el parell $\langle b:B, c:C \rangle$, que és un enllaç binari. Per tant $a:A$ ha de tenir visibilitat del model de l'enllaç binari $\langle b:B, c:C \rangle$.

? (95)

Com expresseu la visibilitat sobre un enllaç binari?

Reducció a un problema conegut. Aquest problema no és nou: volem enllaçar un objecte $a:A$ al model d'un enllaç. Aquest és exactament el problema que calia resoldre en el concepte associatiu: l'objecte associatiu calia enllaçar-lo a l'enllaç pertinent.

Diferències amb el problema conegut. En un concepte associatiu, però, l'objecte associatiu determinava unívocament l'enllaç; i a la inversa. Ara la multiplicitat de l'associació ternària permet que una $a:A$ s'associï a més d'un parell $\langle B, C \rangle$. Per tant $a:A$ no el podem considerar un concepte associatiu entre B i C !

Solució dels problemes. El que cal és considerar el parell $\langle B, C \rangle$ com un concepte associatiu, diguem-ne X , i considerar una associació $M - N$ entre A i X . D'aquesta manera cada $a:A$ s'associa a múltiples parells $\langle b:B, c:C \rangle$.

Transformació a un model equivalent. Per tant, per modelitzar una associació ternària el que cal és transformar el model a un model on tota ternària s'ha modelitzat amb un concepte associatiu i una associació binària. El model transformat és del tot equivalent, ja que l'únic que fem en la transformació és veure cada 3-tuple $\langle a:A, b:B, c:C \rangle$ com un 2-tuple a dos nivells $\langle a:A, \langle b:B, c:C \rangle \rangle$.

(116)

Transformació de les associacions ternàries

Donada una associació ternària $M - N - P$ entre A, B i C , podem transformar-la a un concepte associatiu X entre B i C , i una associació $M - N$ entre A i X . El model resultant és equivalent al de partida.

Capítol 24

GLS La idea de *participació*

1	El concepte <i>Persona</i>	475
1.1	<i>Persona</i> en el MC	475
1.2	El component <i>Persona</i>	475
1.3	Les mancances del model	476
2	El concepte <i>Participant</i>	476
2.1	<i>Participant</i> en el MC	476
2.2	El component <i>Participant</i>	477
3	El concepte <i>Participació</i>	478
3.1	Necessitat del concepte <i>Persona</i>	478
3.2	El model sense el concepte <i>Persona</i>	479
3.3	<i>Participació</i> en el MC	480
3.4	El component <i>Participació</i>	481
3.5	Replantejament del tipus d'identificació	482
3.6	Reintroducció de <i>Persona</i>	483
4	El concepte <i>Línia d'inscripció</i>	483
4.1	Revisió del component <i>Participació</i>	483
4.2	La <i>Línia d'inscripció</i> en el MC	484
4.3	El component <i>Línia d'inscripció</i>	484
4.4	Reintroducció de <i>Persona</i>	485
5	Conclusions sobre la <i>participació</i>	486
5.1	Conclusions sobre l'especificació	486
5.2	Comparació amb les conclusionjs prèvies	487

Contingut detallat del capítol 24

1 El concepte <i>Persona</i>	475
1.1 <i>Persona</i> en el MC	475
Plantejament inicial	475
Associació de participació	475
Tanquem el cicle	475
Restricció semàntica	475
1.2 El component <i>Persona</i>	475
Àmbit d'identificació	475
L'ES <i>nouParticipant</i> (<i>nom</i> , <i>edat</i>)	475
Prohibició de doble inscripció	475
Conseqüència de la PRE	475
Repositoris	475
1.3 Les mancances del model	476
La participació	476
La <i>participació</i> com a concepte	476
2 El concepte <i>Participant</i>	476
2.1 <i>Participant</i> en el MC	476
Concepte associatiu	476
La <i>Inscripció</i>	476
Associacions i restriccions semàntiques	476
Propòsit acomplert	476
2.2 El component <i>Participant</i>	477
2.2.1 Plantejament	477
L'ES <i>nouParticipant</i> (<i>nom</i> , <i>edat</i>)	477
Àmbit d'identificació	477
2.2.2 Modelització del concepte associatiu	477
El paper de la <i>Caminada</i>	477
Modelització del concepte associatiu	477
Identificació doble	477
2.2.3 Modelització del concepte associatiu transformat	478
Transformació de MC	478
El model transformat	478
Quasi-equivalència	478
Introducció de la restricció semàntica	478
3 El concepte <i>Participació</i>	478
3.1 Necessitat del concepte <i>Persona</i>	478
Un model que n'enriqueix un altre	478
Necessitat dels conceptes	478
Possible necessitat	478
El cas de <i>Gastem la sola</i>	478
El paper de la simplicitat	479
Amigabilitat inexistent	479
Supressió de la <i>Persona</i>	479

3.2	El model sense el concepte <i>Persona</i>	479
	El <i>Participant</i> sense <i>Persona</i>	479
	El <i>Participant</i> com un tuple	479
	El tuple disminuït	479
	Identificador compost	480
	El paper del segon nom de l'identificasdor	480
	Del concepte associatiu al concepte feble	480
	El concepte <i>Participació</i>	480
3.3	<i>Participació</i> en el MC	480
	Semàntica de la <i>Participació</i>	480
	Concepte feble	480
	Associacions	481
	Restricció semàntica	481
3.4	El component <i>Participació</i>	481
	Modelització del concepte feble	481
	Identificació externa	481
	Comprovació d'identificador	481
3.5	Replantejament del tipus d'identificació	482
	Identificació externa o interna	482
	Responsabilitats en la identificació externa	482
	Responsabilitats en la identificació interna	482
	Recuperació d'identificador	482
	Violació dels requeriments	482
	Noms sense semàntica	482
	D'identificador a atribut	482
3.6	Reintroducció de <i>Persona</i>	483
	Significat de la reintroducció	483
	Recuperació de l'associativitat	483
	Canvis en els contractes	483
	Pèrdua no traumàtica	483
4	El concepte <i>Línia d'inscripció</i>	483
4.1	Revisió del component <i>Participació</i>	483
	Replantejament de l'àmbit d'identificació	483
	Conseqüències d'un canvi d'àmbit d'identificació	483
	El paper de la <i>Inscripció</i>	484
	El concepte <i>Línia d'inscripció</i>	484
	Restriccions sobre els noms de persona	484
4.2	La <i>Línia d'inscripció</i> en el MC	484
	Feblesa i identificació	484
	Associacions	484
	Inexistència del sícle	484
4.3	El component <i>Línia d'inscripció</i>	484
	Modelització del concepte feble	484
	Aventatges	484
	Inconvenients	484
	Recuperem la semàntica perduda	484

El preu dels inconvenients	485
4.4 Reintroducció de <i>Persona</i>	485
Significat de la reintroducció	485
Recuperació de les comprovacions d'identificador	485
Canvis en els contractes	486
Pèrdua no traumàtica	486
5 Conclusions sobre la <i>participació</i>	486
5.1 Conclusions sobre l'especificació	486
El primer intent	486
Cerquem la simplicitat	486
Un model acceptable	486
Dos models que ens satisfan	486
Simplicitat dels models	486
Dues opcions per considerar	486
5.2 Comparació amb les conclusionjs prèvies	487
Conclusions prèvies	487
<i>Partícep i Línia d'inscripció</i>	487
<i>Partícep i Participació</i>	487
<i>Partícep i Participant</i>	487
Arribem al mateix lloc	487

1 El concepte *Persona*

1.1 *Persona* en el MC

Plantejament inicial. El primer intent de modelització de l'enunciat porta a definir un concepte *Persona*, amb una associació $s'inscriu:M - N$ amb *Inscripció*.

Associació de participació. Per assegurar que una mateixa *persona* no s'inscriu més d'un cop en una mateixa *Caminada* podem pensar en una associació $participa:M - N$ entre *Caminada* i *Persona*, i en una restricció semàntica sobre ella.

Tanquem el cicle. La restricció semàntica necessària ha de fer ús de l'associació $referent-a:1 - N$ entre *Caminada* i *Inscripció*. Les tres associacions considerades formen un cicle:

$$s'inscriu \bullet referent-a = participa$$

Restricció semàntica. La restricció semàntica necessària ha de dir que una mateixa *Persona* pot *inscriure's* a diferents *Inscripcions* sempre i quan aquestes *es refereixin* a diferents *Caminades*.

1.2 El component *Persona*

Àmbit d'identificació. Una mateixa *persona* pot aparèixer en *Caminades* diferents. Això exigeix que l'àmbit d'identificació de *Persona* sigui un àmbit global a totes les *Caminades*.

L'ES *nouParticipant* (*nom*, *edat*). L'ES *nouParticipant* (*nom*, *edat*) té un argument *nom* que és l'identificador de *Persona*. Això exigeix (pel principi de la *Identificació robusta*) un mecanisme de *comprovació d'identificador*: *existeixPersona* (*nom*) : <bool>.

Prohibició de doble inscripció. La restricció semàntica de la no doble inscripció a una mateixa *Caminada* es pot introduir amb una PRE a l'ES *nouParticipant* (*nom*, *edat*) que exigeixi que la *p:Persona* indicada no s'hagi inscrit prèviament a la *c:Caminada* activa.

Conseqüència de la PRE. El fet d'afegir la PRE que la *p:Persona* no s'hagi inscrit prèviament a la *c:Caminada* activa obliga a introduir un nou mecanisme de *comprovació d'identificador*: *existeixPersonaEnCaminada* (*nom*) : <bool>.

Repositoris. L'ES *nouParticipant* (*nom*, *edat*) requereix dos mecanismes de *comprovació d'identificador*: un sobre l'àmbit global; l'altre sobre l'àmbit local de cada *Caminada*. Això exigeix un repositori centralitzat de totes les *Persones* del sistema, i un repositori de *Persones* per *Caminades*. De fet aquest segon repositori és un subconjunt del primer: conté totes les *Persones* que participen en alguna *Caminada*.

1.3 Les mancances del model

La participació. En el MC la *participació* en una *c:Caminada* és una associació entre la *c:Caminada* i la *p:Persona*. Però el model no contempla la possibilitat que la *participació* tingui atributs. Per exemple, l'obsequi que cada *Persona* rep en cada *Caminada*, en record de la seva *participació* on apareix en el model conceptual presentat? I el dorsal? I el temps esmerçat en la fer la *Caminada*?

La *participació* com a concepte. Cal modificar el model per tal que aparegui el concepte de *Participació* o el de *Participant*.

Component <i>Persona</i>	
Comprovació d'identificador	<i>existeixPersona</i> (<i>nom</i>) : <bool> <i>existeixPersonaEnCaminada</i> (<i>nom</i>) : <bool>
Repositoris de <i>Persona</i>	Centralitzat Fragmentat per <i>Caminada</i>
Problemes	No admet atributs sobre la <i>participació</i>

Figura 24.1: Disseny amb el concepte *Persona*

2 El concepte *Participant*

2.1 *Participant* en el MC

Concepte associatiu. Una *p:Persona*, en una *c:Caminada* concreta, esdevé *Participant* d'aquesta *c:Caminada*. Per tant el *Participant* és un concepte associatiu entre *Caminada* i *Persona*.

La *Inscripció*. Una *Inscripció* no enregistra *Persones*, sinó que el que li interessen són les *Persones* en relació a una *Caminada*. Per tant, una *Inscripció* s'ha d'associar amb *Participants*.

Associacions i restriccions semàntiques. El *Participant* té una associació *s'inscriu*: $N - 1$ amb *Inscripció*. Cal però assegurar que tots els *Participants* associats a una mateixa *i:Inscripció* es corresponen a *Participants* d'una mateixa *Caminada*. Per fer-ho podem introduir l'associació *referent-a*: $N - 1$ entre *Inscripció* i *Caminada*, i la restricció de consistència del cicle que formen les tres associacions (*referent-a*, *s'inscriu*, i *Participant* que com a concepte associatiu també és una associació).

Propòsit acomplert. Amb la introducció del concepte de *Participant* hem acomplert el propòsit cercat: ara tenim un espai on fer aparèixer les propietats de cada *participació*. Per exemple, el temps esmerçat en fer una *c:Caminada* concreta per part d'una *p:Persona* determinada, el podem expressar com un atribut del *part:Participant* corresponent; l'obsequi rebut el podem expressar, per exemple, com una associació entre el *part:Participant* i l'*o:Obsequi*.

2.2 El component *Participant*

2.2.1 Plantejament

L'ES *nouParticipant* (*nom*, *edat*). En l'ES *nouParticipant* (*nom*, *edat*) ara el *nom* identifica globalment la *p:Persona*. L'ES té la responsabilitat, entre d'altres, de crear el *part:Participant* corresponent a aquesta *p:Persona* i la *c:Caminada* activa.

Àmbit d'identificació. Per tal que una mateixa *Persona* pugui participar en diferents *Caminades*, però no pugui inscriure's dos cops en la mateixa *c:Caminada*, cal que l'àmbit d'identificació del *part:Participant* sigui la *c:Caminada*.

2.2.2 Modelització del concepte associatiu

El paper de la *Caminada*. La modelització d'un concepte associatiu¹ deixa un grau de llibertat en el paper de cada extrem de l'associació que defineix el concepte associatiu. En aquest cas la restricció exigida pels requeriments fixa el paper de la *Caminada* com el de l'àmbit d'identificació

Modelització del concepte associatiu. De tot plegat en resulta que la *Caminada* té visibilitat d'atribut multiavaluada dels *Participants*; que l'identificador de *Participant* és el de *Persona*; i que el seu àmbit d'identificació és la *Caminada*. La individualització del *Participant* és una individualització induïda.²

Identificació doble. L'argument *nom* de l'ES *nouParticipant* (*nom*, *edat*) és un identificador tant de *Persona* com del *Participant* dins la *Caminada*. Per tant calen dos mecanismes de *comprovació d'identificador*.

Component <i>Participant</i>	
Comprovació d'identificador	<i>existeixPersona</i> (<i>nom</i>) : <bool> <i>existeixPersonaEnCaminada</i> (<i>nom</i>) : <bool> ³
Àmbit d'identificació de <i>Participant</i>	<i>Caminada</i>
Tipus d'identificació	Induïda (per la <i>Persona</i>)
Repositori de <i>Persona</i>	Centralitzat
Repositori de <i>Participant</i>	Fragmentat per <i>Caminada</i>

Figura 24.2: Disseny amb el concepte associatiu *Participant*

¹Vegeu el requadre 108, pàgina 454.

²No es tracta d'una identificació externa perquè qui determina l'identificador és la *Persona*. Tal com hem especificat l'ES *nouParticipant* (*nom*, *edat*) externament indiquem la *p:Persona* amb el seu identificador, la qual cosa exigeix una encaixada inicial. Alternativament haguéssim pogut definir l'ES com *nouParticipant* (*p:Persona*); llavors és més evident que es tracta d'unaq identificació induïda, i no pas externa.

2.2.3 Modelització del concepte associatiu transformat

Transformació de MC. Com hem vist per modelitzar una concepte associatiu podem usar el model transformat quasi-equivalent en el que s'ha suprimit el concepte associatiu.

El model transformat. En el cas que ens ocupa, obtenim un component *Participant* amb una associació $N - 1$ amb *Persona*, i una altra associació $N - 1$ amb *Caminada*. Ara però *Participant* té un identificador propi, no necessàriament induït per la $p:Persona$ sobre la que té visibilitat.

Quasi-equivalència. El model obtingut no és equivalent al model anterior. En concret, admet que *Participants* diferents, puguin referir-se al mateix parell $\langle p:Persona, c:Caminada \rangle$; i per tant permet que una mateixa $p:Persona$ tingui *Inscripcions* diferents sobre una mateixa $c:Caminada$.

Introducció de la restricció semàntica. Si introduïm la restricció semàntica per tal que el model inicial i el transformat siguin equivalents, obtenim la modelització anterior.⁴

3 El concepte *Participació*

3.1 Necessitat del concepte *Persona*

Un model que n'enriqueix un altre. El MC amb el concepte *Participant* es pot veure com un enriquiment del MC amb el concepte *Persona*: la *Persona*, com a concepte, es manté, però a més s'introdueix el concepte *Participant*.

Necessitat dels conceptes. El concepte *Participant* apareix davant la necessitat de modelar atributs sobre la *participació*, con el temps esmerçat en fer la $c:Caminada$, o l'obsequi obtingut en finalitzar-la. Quina és però la necessitat del concepte *Persona*?

? (96)

A efectes del problema, cal el concepte *Persona*?

Possible necessitat. Si de les *persones* que participen en les *caminades* en demanem diverses dades, que es volen persistents, com els noms, adreça, telèfons, etc. és evident la necessitat del concepte *Persona*,

³De fet respon la qüestió de si existeix un **Participant** amb aquest nom en la *Caminada* activa.

⁴Vegeu la taula 24.2.2.2, pàgina 477.

El cas de *Gastem la sola*. Les dades que *Gastem la sola* vol enregistrar són ben minses. I enlloc es demana la persistència d'aquestes dades. Per tant en principi sembla que podríem prescindir del concepte *Persona*.

Si usem un concepte, útil en alguns casos, però sense utilitat en el cas concret, passa alguna cosa?

(97) ?

El paper de la simplicitat. Una de les claus del disseny és obtenir la simplicitat. I aquesta generalment no s'obté si introduïm possibles funcionalitats que ningú no ens ha demanat, encara. És preferible dissenyar allò que ens demanen, de manera simple, mantenible i modificable, que no pas dissenyar un sistema molt complex del qual només ens en demanen una part.

Amigabilitat inexistent. En el cas de *Gastem la sola* l'ús del concepte *Persona* dificulta enormement la tasca de l'usuari: per introduir una participació primer de tot cal comprovar que la *p:persona* que participa ja està en el sistema, i en cas contrari crear-la; un cop assegurada l'existència de la *p:Persona*, cal recuperar-la o encaixar-la; i amb ella s'ha de crear la *part:Participant*. No es tracta només d'un problema de complicació en el disseny, sinó d'interfície d'usuari: primer caldrà assegurar l'existència de la *p:Persona* (amb una cerca sobre la interfície, o amb un formulari de creació), i després es podrà demanar la creació del *part:Participant*. En conclusió, oferim a l'usuari un sistema feixuc, gens amigable.

Supressió de la *Persona*. Concloem per tant, que en el cas que ens ocupa, el concepte *Persona* fa més nosa que servei. Per tant, el suprimim!!!

3.2 El model sense el concepte *Persona*

El *Participant* sense *Persona*. El concepte *Participant* era un concepte associatiu entre *Persona* i *Caminada*. Ara, en suprimir el concepte *Persona*, què en queda?

El *Participant* sense la *Persona* què és? Té sentit?

(98) ?

El *Participant* com un tuple. Tot concepte associatiu el podem veure com un 2-tuple que aparella dos conceptes. El tuple resultant és també un objecte, amb els seus propis atributs. Així tot *part:Participant* es pot veure com un tuple $\langle p:Persona, c:Caminada \rangle$ que s'identifica amb el parell de noms $\langle ID_{Persona}, ID_{Caminada} \rangle$, i que té els seus propis atributs (diguem-ne *atr_{Part}*).

El tuple disminuït. En suprimir la *Persona* el *part:Participant* agrupa un sol objecte (la *Caminada*); però continua identificant-se amb el parell de noms $\langle ID_{Persona}, ID_{Caminada} \rangle$, i manté els seus propis atributs atr_{Part} .

Identificador compost. Així, tot i suprimir el concepte *Persona* el *Participant* continua necessitant un parell de noms per ser identificat: un dels noms identifica també una *c:Caminada*, però l'altre nom no identifica cap concepte del model.

El paper del segon nom de l'identificador. Un canvi en el segon nom, canvia la realització del *Participant* sense modificar la *c:Caminada* a la que aquest està associat. A més, a tots els efectes aquest segon nom ara no té cap mena de significat, i per tant res no impedeix repetir-lo entre *Caminades*.

Del concepte associatiu al concepte feble. El resultat, per tant, és que en suprimir el concepte *Persona* el concepte associatiu *Participant* s'ha convertit en un concepte feble de *Caminada*:

(118)(117)

Afebliment d'un concepte associatiu

En suprimir un dels extrems d'un concepte associatiu aquest es converteix en un concepte feble de l'extrem que sobreviu

El concepte *Participació*. El concepte *Participant* sembla estrany que sigui un concepte feble respecte de la *Caminada*. Estem dient que els *participants* no poden existir sense la *c:Caminada* a la que participen. Segons què entenem per "participant" això és força estrany.⁵ Per això és preferible cercar un nom més adient al concepte que estem manipulant. I aquest nom podria ser el de *Participació*: no es tracta d'un subjecte (com era *Participant*) sinó d'una actuació. El concepte *Participació* es refereix al fet que algú fa alguna cosa; en concret, que algú participa en una *c:Caminada*. I això és independent de si el sistema contempla la *Persona* com a concepte o no.

3.3 *Participació* en el MC

Semàntica de la *Participació*. La *Participació* és el concepte que es refereix a una actuació: al fet d'inscriure's o realitzar una *Caminada*.⁶

⁵Si el *participant* és la *persona* que fa o s'inscriu en una *caminada*, com podem expressar un *participant* sense la *persona* o sense la *caminada*?

⁶No es tracta d'un concepte substantiu com era *Persona* o com és *Caminada*, sinó de la substantivació d'un verb. Aquestes substantivacions són força habituals, tot i que sovint l'acció substantivada es pot assimilar a algun tipus de document o registre de l'acció. Per exemple, en una acció de *compra* s'enregistra la transacció en un *albarà*; per tant és possible considerar el concepte *Compra* com el mateix concepte que *Albarà*. Un altre exemple de substantivació és la *Inscripció*, que és el registre de l'acció d'inscriure's.

Concepte feble. La *Participació* és un concepte feble respecte la *Caminada*. L'identificador dins la *Caminada* és un nom que es pot referir a una persona. (Aquí cal tenir present que “persona” no és un concepte de MC, i que per tant aquesta referència només té sentit fora del sistema. A més, com que no té cap rellevància de cara al sistema, aquesta referència no és necessària; d'aquí el “pot”).

Associacions. La *Participació* té una associació *s'inscriu*: $N-1$ ⁷ amb *Inscripció*; i aquesta una associació *referent-a* amb *Caminada*.

Restricció semàntica. Com introduïm la restricció semàntica que no hi poden haver dues *Inscripcions* diferents sobre la mateixa *c:Caminada* on hi participi la mateixa *p:persona* si la *Persona* no apareix en el MC? Si assumim que la idea de *persona* queda oculta en el nom que fa d'identificador de la *Participació* dins de la *Caminada* llavors és evident que la repetició d'aquest nom (és a dir, de la *persona*) només es pot donar en *Caminades* diferents.

3.4 El component *Participació*

Modelització del concepte feble. La modelització del concepte feble⁸ exigeix un repositori de *Participacions* fragmentat per *Caminades*, on cada fragment és un àmbit d'identificació independent.

Identificació externa. El nom de les persones l'usem com a identificador de la *Participació* dins la *Caminada*. Es tracta per tant d'una *identificació externa*: el valor d'identificació és generat fora del sistema; el sistema no hi realitza cap comprovació. (Les úniques restriccions sobre el valor d'identificació apareixen en els contractes dels ES).

Comprovació d'identificador. El nom de les persones és l'identificador de la *Participació* dins la *Caminada*. Per tant l'ES *nouParticipant* (*nom*, *edat*) rep un argument que és un identificador, la qual cosa exigeix un mecanisme de *comprovació d'identificador*: *existeixNomEnCaminada* (*nom*) : <bool>.

Component <i>Participació</i>	
Comprovació d'identificador	<i>existeixNomEnCaminada</i> (<i>nom</i>) : <bool>
Àmbit d'identificació de <i>Participació</i>	<i>Caminada</i>
Tipus d'identificació	Externa
Repositori de <i>Participació</i>	Fragmentat per <i>Caminada</i>

Figura 24.3: Disseny amb el concepte feble *Participació*

⁷Ara el nom de l'associació no és massa afortunat. Si usem *referent-a* es confon amb una altra associació; si usem *participació* un mateix nom expressa un concepte i una associació.

⁸Vegeu el requadre 115, pàgina 468.

3.5 Replantejament del tipus d'identificació

Identificació externa o interna. La modelització d'un concepte feble ofereix totes les condicions per a la individualització genètica. Tot i així, estem usant la identificació externa per a la *Participació*. Què és el que realment ens interessa?

Responsabilitats en la identificació externa. La *identificació externa* traspasa la responsabilitat de crear els valors d'identificació a l'usuari del sistema. En el cas que ens ocupa això no és problemàtic perquè estem assumint que aquests valors són els noms de persones reals. La contrapartida és que el sistema ha d'oferir el mecanisme de comprovació d'identificador.

Responsabilitats en la identificació interna. En el cas d'usar la *identificació externa* per a la *Participació*, el nom rebut com argument amb l'ES `nouParticipant (nom, edat)` esdevé part de la informació d'expertesa en la creació. I per tant desapareix la necessitat del mecanisme de comprovació d'identificador! El preu a pagar és que ara no podem impedir noms idèntics en *Participacions* diferents dins d'una mateixa `c:Caminada`, i que caldrà introduir algun mecanisme de recuperació de l'identificador de la `part:Participació` generat internament.

? (99) Quin mecanisme de recuperació de l'identificador de *Participació* cal, en el cas d'identificació interna?

Recuperació d'identificador. El mecanisme més simple és que el valor d'identificació s'obtingui com a valor de retorn de l'ES que demana la creació:
`nouParticipant (nom, edat) : <Id>`.

? (100) La repetició de noms dins les *Participacions* d'una mateixa `c:Caminada` és un problema greu?

Violació dels requeriments. Sembla obvi que la repetició de noms de persona en diferents *Participacions* d'una mateixa `c:Caminada` viola els requeriments. Malgrat tot aquesta violació podem defensar que és només aparent: per considerar-la una autèntica violació caldria que MC continguéss el concepte *Persona*. Com un model que no té el concepte *Persona* pot violar propietats sobre les *persones* (en concret la no doble *inscripció* en una mateixa `c:Caminada`)??

Noms sense semàntica. De fet, en suprimir el concepte *Persona*, els noms de les *persones* han perdut tota la semàntica dins del sistema, i per tant perden el seu poder comunicatiu entre el sistema i l'exterior: el *nom* té sentit per l'usuari, però no en té pel sistema. Llavors, perquè volem exigir que el *nom* d'una persona identifiqui una *Participació*?

D'identificador a atribut. Suposem que el *nom de la persona* és un atribut més de la *Participació*, però no n'és pas el seu identificador. Llavors deixem la porta oberta a la *identificació interna*, però traspassem a l'usuari tot possible control de consistència sobre els noms.

Component <i>Participació</i> (2)	
Àmbit d'identificació de <i>Participació</i>	Caminada
Tipus d'identificació	Genètica (El creador és la <i>Caminada</i>)
Repositori de <i>Participació</i>	Fragmentat per <i>Caminada</i>

Figura 24.4: Disseny amb el concepte feble *Participació*

3.6 Reintroducció de *Persona*

Significat de la reintroducció. Un cop desenvolupat el sistema amb el concepte *Participació* no és massa complex reintroduir el concepte de *Persona*. L'únic que cal és recuperar la semàntica del *nom*; és a dir, assegurar que a part de totes les funcions que té assignades també sigui identificador de *Persona*.

Recuperació de l'associativitat. El que cal fer és afegir el concepte *Persona*, associar-lo a *Participació*, i afegir les restriccions semàntiques necessàries per tal que la *Participació* es comporti com un concepte associatiu entre *Caminada* i *Persona*.

Canvis en els contractes. L'aparició del concepte *Persona* i la recuperació de la semàntica del *nom* com a identificador de *Persona* obliga a modificar els contractes dels ES; i en el disseny a introduir l'encaixada.

Pèrdua no traumàtica. En conclusió, l'ús del concepte *Participació* no comporta una pèrdua considerable: en qualsevol moment que es cregui convenient es pot reintroduir el concepte *Persona* de manera no traumàtica.

4 El concepte *Línia d'inscripció*

4.1 Revisió del component *Participació*

Replantejament de l'àmbit d'identificació. Si els *noms de persona* no tenen cap significat pel sistema, fins el punt que no s'ofereixen mecanismes per controlar la seva unicitat en les *Caminades*, quin sentit té mantenir la *Caminada* com a àmbit d'identificació de la *Participació*? És aquest l'àmbit d'identificació més idoni?

Conseqüències d'un canvi d'àmbit d'identificació. Si canviem l'àmbit d'identificació de la *Participació*, llavors aquesta deixa de ser un concepte feble respecte la *Caminada*. I caldrà canviar-li el nom.

? (101)

Quan apareix una *participació* o un *participant*? Com apareix?

El paper de la *Inscripció*. És l'ES *nouParticipant* (*nom*, *edat*) qui introdueix els participants o les participacions en el sistema. El contracte d'aquest ES demana una *i:Inscripció* activa. Per tant, el *participant* o la *participació* és un concepte lligat a la *Inscripció*, més que no pas a la *Caminada*.

El concepte *Línia d'inscripció*. En conseqüència, podem considerar un concepte *Línia d'inscripció* com l'element constitutiu bàsic d'una *Inscripció*. Cada *Línia d'inscripció* té un atribut amb el *nom d'una persona*.

Restriccions sobre els noms de persona. És responsabilitat de l'usuari assegurar la consistència dels noms de les diferents *Línies d'inscripció*.

4.2 La *Línia d'inscripció* en el MC

Feblesa i identificació. La *Línia d'inscripció* és un concepte feble respecte la *Inscripció*. El *nom* de les *Línies* és un atribut no identificador. La identificació de la *Línia* és interna.

Associacions. La *Caminada* té una associació *referent-a:1 – N* amb *Inscripció*; la *Inscripció* té una associació *formada-per:1 – N* amb *Línia*.

Inexistència del cicle. Ara no hi ha cap necessitat d'una associació entre *Caminada* i *Línia d'inscripció*. Per tant el cicle de les associacions ha desaparegut, i amb ell la necessitat de restriccions semàntiques per mantenir la consistència del cicle.

4.3 El component *Línia d'inscripció*

Modelització del concepte feble. L'àmbit d'identificació de la *Línia* és la *Inscripció*; el repositori de *Línies* és fregmentat per *Inscripció*; i s'usa identificació interna per a la *Línia*. Desapareix la necessitat de la comprovació d'identificador.

Aventatges. L'aventatge principal d'usar el component *Línia d'inscripció* és la simplicitat del model: desapareixen els cicles entre les associacions; totes les associacions esdevenen *1 – N*; i desapareix la necessitat de la comprovació d'identificador.

Inconvenients. En usar el component *Línia d'inscripció*, l'inconvenient principal és la pèrdua total de la semàntica dels noms de les persones. Com a conseqüència no té sentit exigir condicions d'unicitat dels noms en determinats contextos; la qual cosa porta a una possible violació dels requeriments.

Recuperem la semàntica perduda. Part de la semàntica perduda amb l'ús de la *Línia d'inscripció* pot ser recuperada a través de la capa de *Presentació*. Per exemple, si les *Inscripcions* s'introdueixen mitjançant un formulari tabular amb un botó de confirmació final, és ben simple exigir que no hi hagi files en el formulari amb el mateix valor a la columna *nom*. En canvi, per exemple, la unicatat del *nom* en la *Caminada* no és possible només amb la *Presentació*.

El preu dels inconvenients. En cada cas caldrà estudiar si la simplicitat obtinguda amb la pèrdua de la semàntica dels noms compensa o no la possible pèrdua de requeriments. En el cas que ens ocupa, per exemple, veiem que la *Presentació* pot introduir mecanismes per evitar errades en les noves *Inscripcions*; i podem assumir que és responsabilitat de l'usuari que es vol inscriure saber si ho ha fet abans o no.

Exemple 106 (Requeriments excessius) *Té sentit que un sistema de reserva de vols comprovi si el comprador d'un vol ja n'ha comprat un altre, per qualsevol altra destinació i per qualsevol companyia, però pel mateix dia?*

El cost de procés per ressaltar aquest control és clarament excessiu. En canvi filtrariem pocs casos: qui és el que s'oblida que té un bitllet per Bangkok quan compra el bitllet per anar a Ulan Bator?

Component <i>Línia d'inscripció</i>	
Àmbit d'identificació de <i>Línia</i>	<i>Inscripció</i>
Tipus d'identificació	Genètica (el creador és la <i>Inscripció</i>)
Repositori de <i>Línies</i>	Fragmentat per <i>Inscripció</i>
Aventatge	Simplicitat: no hi ha cicles; associacions $1 - N$
Inconvenient	Cap control sobre els noms, tot i que la <i>Presentació</i> en pot reintroduir algun (però no tots)

Figura 24.5: Disseny amb el concepte feble *Línia d'inscripció*

4.4 Reintroducció de *Persona*

Significat de la reintroducció. Un cop desenvolupat el sistema amb el concepte *Línia d'inscripció* no és massa complex reintroduir el concepte de *Persona*. L'únic que cal és recuperar la semàntica del *nom*; és a dir, assegurar que a part de totes les funcions que té assignades també sigui identificador de *Persona*.

Recuperació de les comprovacions d'identificador. El que cal fer és afegir el concepte *Persona* i associar-lo a la *Línia d'inscripció*. En el disseny caldrà replicar el repositori de *Persones* en els fragments on volgüem unicatat, i per tant un mecanisme de comprovació d'identificador. Per exemple, caldrà un repositori de *Persones* fragmentat pe *Caminades* per permetre les compro-

vacions que assegurin que una mateixa *p:Persona* no s'inscriu dos cops en la mateixa *c:Caminada*.

Canvis en els contractes. L'aparició del concepte *Persona* i la recuperació de la semàntica del *nom* com a identificador de *Persona* obliga a modificar els contractes dels ES; i en el disseny a introduir l'encaixada.

Pèrdua no traumàtica. En conclusió, l'ús del concepte *Línia* no comporta una pèrdua considerable: en qualsevol moment que es cregui convenient es pot reintroduir el concepte *Persona* de manera no traumàtica.

5 Conclusions sobre la *participació*

5.1 Conclusions sobre l'especificació

El primer intent. L'especificació directa de l'enunciat porta a la necessitat d'un concepte *Participant*. El MC resultant és feixuc, i dificulta l'amigabilitat del sistema que es vol construir.

Cerquem la simplicitat. Una anàlisi atenta del problema en concret porta a l'observació que el concepte de *Persona* és innecessari. Els canvis que comporta la supressió de la *Persona* obliguen a bandejar el concepte *Participant* i a introduir la *Participació* en el seu lloc.

Un model acceptable. El model basat en el concepte *Participació* compleix amb tots els requeriments, però en alguns aspectes ens deixa insatsfets. En concret tot el disseny es basa en un element no especular: internament volem mantenir el *nom* de les *Persones*, però a MC no existeix el concepte *Persona*.

Dos models que ens satisfan. Si suprimim tota semàntica associada al *nom* de les *persones*, obrim la porta a la *identificació interna*. Llavors podem usar el concepte *Participació*, amb la *Caminada* com a àmbit d'identificació de *Participació*; o podem substituir el concepte *Participació* pel concepte *Línia d'inscripció*, i usar la *Inscripció* com a àmbit d'identificació de *Línia*.

Simplicitat dels models. Els models resultants són altament simple, però són incapaç de mantenir les restriccions que els requeriments exigeixen sobre les *persones*. Malgrat tot, l'ús de la *Presentació* permet recuperar part de la pèrdua; i allò que no és recuperable podem assumir que és un excés en els requeriments.


Dues opcions per considerar. En conclusió, admetem com a vàlids tant el model basat en la *Participació* com el basat en la *Línia d'inscripció*, cadascun amb els seus pros i contres. I refusem el model basat en el *Participant* per massa feixuc. En qualsevol cas, sigui quin sigui el model emprat, sempre podem reintroduir el concepte *Persona* de manera no traumàtica.

5.2 Comparació amb les conclusionjs prèvies

Conclusions prèvies. Al capítol 22., pàgina 428, es partia d'un MC fixat, amb el concepte *Partícep*. El contracte de l'ES *nouParticipant* (*nom*, *edat*) tenia una llacuna que hem intentat omplir amb el desenvolupament àgil. La conclusió a la que arribàvem és que calia identificació interna, i àmbit d'identificació o bé global o bé el de la *Inscripció*.

***Partícep* i *Línia d'inscripció*.** El MC del capítol 22 coincideix amb el MC obtingut aquí amb el concepte *Línia d'inscripció*: allà on teníem *Partícep* ara tenim la *Línia d'inscripció*. Les conclusions sobre el disseny també convergeixen: usem *identificació interna* de la *Línia d'inscripció*, i la *Inscripció* com el seu àmbit d'identificació.

Usem el concepte *Línia d'inscripció*. En el disseny usem identificació interna per a la *Línia d'inscripció*, i la *Inscripció* com el seu àmbit d'identificació.

(102) 

***Partícep* i *Participació*.** En el capítol 22. hem bandejat la *Caminada* com a àmbit d'identificació de *Partícep* perquè comportava la violació del principi de la *Consistència individualitzadora*. Aquí però, com que tenim permís per modificar el MC, obtenim una proposta vàlida, basada en el concepte *Participació*, on la *Caminada* és l'àmbit d'identificació de la *Participació*.

***Partícep* i *Participant*.** En el capítol 22. hem partit d'un MC on no hi havia *Persona*. Aquí, en canvi, hem analitzat les conseqüències de tenir el concepte *Persona*.

Arribem al mateix lloc. Per tant, podem concloure que les conclusions són les mateixes. La diferència és que en el capítol 22., pàgina 428, teníem el MC fixat, i per tant un grau de llibertat menys.

Capítol 25

GLS CU *ferInscripció* (2)

1	Punt de partida	492
1.1	L'especificació de partida	492
1.2	Model construït fins el moment	492
2	Controlador	493
3	POST: Creació de <i>li</i>:Línia	493
3.1	Anàlisi dels diferents <i>creadors</i>	493
3.2	La <i>Inscripció</i> com a creadora de la Línia	494
3.3	El <i>Controlador</i> com a creador orb no canònic ¹	496
3.4	La <i>Caminada</i> com a creador orb no canònic	498
3.5	Identificació en el cas dels creadors orbs	498
3.6	Comparació de les alternatives	499
4	POST: Enllaç entre <i>i</i> i <i>li</i>; <i>i</i> activa	501
5	Esdeveniment <i>fiInscripcio</i>()	501
5.1	Contracte <i>fiInscripcio</i> ()	501
5.2	POST: Cap <i>Inscripció</i> activa	501

¹Recordem que l'anàlisi dels *creadors orbs* la presentem per completesa expositiva.

Contingut detallat del capítol 25

1 Punt de partida	492
1.1 L'especificació de partida	492
1.2 Model construït fins el moment	492
Components detectats	492
Creador d'Inscripció	492
Recuperació d'informació	492
Inscripció activa	492
Caminada activa	492
Identificadors	492
Identificació de Línia d'inscripció	493
El nom no és un identificador	493
2 Controlador	493
Comunicació entre ES	493
Controlador de CU	493
3 POST: Creació de la Línia	493
3.1 Anàlisi dels diferents creadors	493
Creació amb emmagatzematge	493
Anàlisi dels creadors orbs	493
3.1.1 Creadors canònics	494
Informació d'expertesa	494
Tipus d'informació d'expertesa	494
Expert	494
Experts per delegació	494
Emmagatzemadors canònics	494
Creador canònic	494
3.2 La Inscripció com a creadora de la Línia	494
Visibilitat d'emmagatzematge	494
Cadena de delegacions	494
Repositori d'Línies	494
Recuperació d'informació	495
Àmbit d'identificació	495
Avaluació de l'alternativa	495
Validesa de les decisions	495
Interpretació de les hipòtesis	495
3.3 El Controlador com a creador orb no canònic ²	496
Aspectes bàsics del model	496
Recuperació de la Inscripció	496
Acoblaments en la recuperació de la Inscripció	496
Visibilitat sobre la Inscripció	496
Recuperació de la Caminada	497
Visibilitat d'Inscripció sobre Caminada	497

²Recordem que l'anàlisi dels *creadors orbs* la presentem per completesa expositiva.

Visibilitat de Línia sobre Caminada	497
Acoblaments en la recuperació de la Caminada	497
Rèplica del repositori d'Inscripcions	497
Repositori de Caminades	497
Superficialitat de l'anàlisi	497
3.4 La Caminada com a creador orb no canònic	498
Aspectes bàsics del model	498
Recuperació de la Inscripció	498
Acoblaments en la recuperació de la Inscripció	498
Repositori d'Inscripcions	498
Superficialitat de l'anàlisi	498
3.5 Identificació en el cas dels creadors orbs	498
Els principis sobre identificació	498
El controlador: creador i individualitzador sobre Caminada	499
El controlador: creador i individualitzador sobre Inscripció	499
La Caminada com a creadora i individualitzadora	499
3.6 Comparació de les alternatives	499
4 POST: Enllaç entre i i li; i activa	501
Enllaç entre i i li	501
Manteniment de i:Inscripció activa	501
5 Esdeveniment fiInscripcio()	501
5.1 Contracte fiInscripcio()	501
5.2 POST: Cap Inscripció activa	501
Objectiu a assolir	501
Modificació de l'atribut pertinent de c:Caminada	501
Assignació a l'atribut InsActiva del valor null	501
Caminada activa sense Inscripció activa	501
Necessitat de desactivar la c:Caminada	502
Desactivació de la c:Caminada	502
Necessitat de desactivar la i:Inscripció	502

1 Punt de partida

1.1 L'especificació de partida

Partim del MC revisat amb el concepte *Línia d'inscripció*, i el contracte de *nouParticipant* (*nom*, *edat*) modificat convenientment.

Contracte (Amb el concepte <i>Línia</i>)	
Paràmetres	
• <i>nom</i> :	Nom del participant
• <i>edat</i> :	Edat del participant
PRE	
1.	Hi ha una <i>i</i> : <i>Inscripció</i> activa
POST	
1.	S'ha creat una nova realització <i>li</i> : <i>Línia</i> ^a
2.	S'ha creat un enllaç entre <i>i</i> i <i>li</i> , corresponent a l'associació <i>s'inscriu</i>
3.	La inscripció <i>i</i> es manté activa
<hr/>	
^a La <i>Línia</i> és de fet <i>Línia d'inscripció</i> .	

1.2 Model construït fins el moment

Components detectats. Partim del MComp construït fins el moment, que conté GLS, *Caminada* i *Inscripció*. Per tant en el moment d'assignar qual-sevol responsabilitat primer de tot mirarem si algun d'aquests tres components pot rebre-la.

Creador d'*Inscripció*. Usem la *Caminada* com a creadora canònica d'*Inscripció*. El resultat és un repositori d'*Inscripcions* fragmentat per *Caminades*.

Recuperació d'informació. Assumim que a les *Inscripcions* s'hi accedeix sempre a través de la *c*:*Caminada* pertinent. Per tant no ens cal cap visibilitat d'atribut per recuperar la *c*:*Caminada* a la que "es refereix" una *i*:*Inscripció*.

Inscripció activa. L'existència d'una *i*:*Inscripció* activa l'hem modelitzada amb una visibilitat indirecta des del controlador passant per *Caminada*. És a dir, el controlador té visibilitat d'atribut de *c*:*Caminada*; i aquesta d'*i*:*Inscripció*:
 $GLS \xrightarrow{atr} Caminada + Caminada \xrightarrow{atr} Inscripció.$

Caminada activa. La visibilitat indirecta des del controlador de la *i*:*Inscripció* activa té com a conseqüència que sempre que tinguem una *i*:*Inscripció* activa també tenim activa la *c*:*Caminada* corresponent.

Identificadors. L'ES *novaInscripcio(c:Caminada)* rep el nom de la *c:Caminada* com a argument. Aquest nom és un identificador, i per tant cal un mecanisme de comprovació d'identificador sobre el nom de la *Caminada*. Per altra banda, la *Inscripció* és d'identificació interna, i per tan no requereix cap mecanisme de comprovació d'identificador.

Identificació de Línia d'inscripció. Cal usar identificació interna. Com a àmbits d'identificació podem considerar l'àmbit global o el local a *Inscripció*. L'àmbit d'identificació local a *Caminada* l'hem descartat per violar el principi de la *Consistència individualitzadora*.


El nom no és un identificador. El nom de persona rebut com a argument en l'ES *nouParticipant(nom, edat)* no és cap identificador. Per tant no cal cap mecanisme de comprovació d'identificador sobre ell.

2 Controlador

Comunicació entre ES. L'exigència d'una *Inscripció* activa en una de les PRE obliga a tenir un mecanisme de comunicació entre ES; en concret, entre l'ES *novaInscripcio(c:Caminada)*, que activa una inscripció, i l'ES *nouParticipant(nom, edat)*, que pren la *Inscripció* activa com la *Inscripció* sobre la que ha de treballar.

Controlador de CU. La manera d'aconseguir aquesta comunicació és aplicant el principi del *Controlador de CU*: usem el mateix *controlador* que els altres ES del mateix CU.

Prenem com a *controlador del CU* a *gls:GLS*

(103) 

3 POST: Creació de *li:Línia*

3.1 Anàlisi dels diferents *creadors*

Creació amb emmagatzematge. Estem davant d'una creació amb emmagatzematge. Per tant, segons el principi *Creador canònic*, cal cercar els experts que siguin emmagatzemadors canònics, i atorgar-los el paper de *creador*.

Anàlisi dels creadors orbs. L'ús de creadors no canònics està contemplat pel principi *Creador orb*, que com hem vist només s'aplica sota circumstàncies excepcionals; en concret, quan els creadors canònics no ens donin un disseny convincent. Tot i no ser aquest el cas, però, per completesa expositiva, després de considerar els creadors canònics analitzarem el disseny resultant en el cas d'usar creadors no canònics.

3.1.1 Creadors canònics

Informació d'expertesa. Per crear una *Línia* cal tant la informació rebuda com a argument de l'ES, com la *Inscripció* a la que pertany la *Línia* en qüestió.

Informació d'expertesa interna i externa

Tipus d'informació d'expertesa. A la informació rebuda com a argument de l'ES i que és necessària per a la creació d'una nova realització (el nom i l'edat en el nostre cas) l'anomenarem *informació externa d'expertesa*; la informació necessària per a la creació que cal recuperar de dins el sistema l'anomenarem *informació interna d'expertesa* (la *Inscripció* en el cas que ens ocupa).

Expert. El controlador és l'*expert*: té visibilitat de paràmetre de la informació externa d'expertesa, i visibilitat indirecta monoavaluada de la informació interna d'expertesa.

Experts per delegació. La *visibilitat d'activació* assegura l'existència d'un enllaç dirigit d'atribut $gls:GLS \rightarrow i:Inscripció$ que manté tant una *c:Caminada* activa com una *i:Inscripció*. Per tant, l'expertesa del controlador es pot transmetre, a través d'aquest enllaç dirigit, tant a la *c:Caminada* com a *i:Inscripció*. De fet només cal transmetre la informació externa d'expertesa.

Emmagatzemadors canònics. En el MC que estem considerant, l'única associació que hi ha amb *Línia* és l'associació *s'inscriu* entre aquesta i *Inscripció*. Per tant, *Inscripció* és l'únic *emmagatzemador canònic*.

Creador canònic. L'únic *expert* (en aquest cas *per delegació*) que també és *emmagatzemador canònic* és *Inscripció*.³

➤ (104)

Prenem com a *creador* de la *Línia* la *Inscripció*

3.2 La Inscripció com a creadora de la Línia

Visibilitat d'emmagatzematge. La visibilitat d'atribut exigida per l'emmagatzematge és, per tant, $Inscripció \xrightarrow{atr} Línia^*$, que es correspon al model de l'associació *s'inscriu*.

Cadena de delegacions. El controlador rep la petició; i la *i:Inscripció* activa la realitza. Per tant cal que el controlador delegui en *c:Caminada* la creació de la *li:Línia*; al seu torn *c:Caminada* delega aquesta creació en la *i:Inscripció*. La delegació és possible gràcies a la *visibilitat d'activació* $GLS \rightarrow Caminada \rightarrow Inscripció$.


³Tot i que no és cap sorpresa, paga la pena remarcar-ho: el principi de la *Consistència individualitzadora* és una conseqüència del principi de la *Individualització efectiva* i del *Creador canònic*. Per això obtenim els mateixos resultats: l'individualitzador de *Línia* ha de ser la *Inscripció* i no pas la *Caminada*.

Repositori d'Línies. El fet que la *Inscripció* sigui la responsable de crear i emmagatzemar les *Línies* significa que cada *i:Inscripció* emmagatzema les seves *Línies*. Amb altres paraules, estem assignant a cada *i:Inscripció* la responsabilitat de mantenir un repositori amb les seves *Línies*; en conseqüència, tenim un *repositori distribuït* de *Línies* on cada *Inscripció* en conté un fragment.⁴

Recuperació d'informació. Si assumim que a les *Línies* sempre s'hi accedeix a partir de la *i:Inscripció* apropiada; i a aquesta sempre s'hi accedeix a través de la *c:Caminada* pertinent, llavors no cal mantenir cap tipus de visibilitat d'atribut de *Línia* cap a *Inscripció* o cap a *Caminada*.

Àmbit d'identificació. Tot plegat és consistent amb el punt de partida, i més concretament, amb la identificació interna de la *Línia*. Ara però veiem que el repositori fragmentat de *Línies* per *Inscripció* exigeix⁵ que l'àmbit d'identificació de *Línia* sigui local a la *Inscripció*.

L'àmbit d'identificació de la *Línia* és la *Inscripció*; la identificació és interna.

(105) 

Avaluació de l'alternativa. La delegació de la responsabilitat de creació de la *Línia* significa que tant el controlador com *Caminada* no estan acoblats a *Línia*. En aquesta delegació, el controlador, que és l'*expert*, transmet la seva expertesa (externa) a través dels atributs dels missatges de delegació. D'aquesta manera el controlador és fidel al principi de l'*Encarrilament*. Per contra, per mor de la delegació, tant *Caminada* com *Inscripció* queden acoblades a la informació externa d'expertesa.

Validesa de les decisions. Les decisions preses (en concret, l'ús del creador canònic) es basen en els supòsits o hipòtesis següents, que caldrà validar al llarg de tot el disseny (segons el principi de la *Reconsideració*):

1. En el MC no hi ha cap associació entre *GLS* o *Caminada* i *Inscripció*
2. En cap moment necessitem un repositori centralitzat de les *Línies*
3. En cap moment necessitem un repositori de les *Línies* distribuït on cada *Caminada* en sigui un fragment

(106) H

⁴Res no impedeix, però, que aquest repositori distribuït sigui una rèplica lògica d'un repositori centralitzat.

⁵Assumim que no hi ha cap rèplica centralitzada del repositori de *Línies*.

Interpretació de les hipòtesis. La primera hipòtesi impedeix considerar el controlador o la Caminada com a *creador canònic*. Les següents hipòtesis diuen que a les Línies sempre hi accedirem des de la Inscripció pertinent, mai des del controlador (segona hipòtesi) o des de la Caminada (tercera hipòtesi). A més, la primera hipòtesi impedeix considerar segons el principi de la *Consistència individualitzadora*, que l'àmbit d'identificació del Línia sigui el controlador o la Caminada; la segona hipòtesi diu que la l'àmbit d'identificació de la Línia no cal que sigui global, mentre que la tercera hipòtesi diu que no cal que sigui la Caminada.⁶

3.3 El Controlador com a creador orb no canònic⁷

Aspectes bàsics del model. Assignar la responsabilitat de creació de la Línia al controlador viola el principi de l'*Encarrilament* i *acobla* el controlador amb Línia. El resultat és un repositori centralitzat de Línies. Per altra banda s'introdueix una visibilitat d'atribut $GLS^{atr} \rightarrow Línia$ que no es correspon a la modelització de cap associació de MC.

Recuperació de la Inscripció. En el cas d'un repositori centralitzat de Línies, per conèixer a quina Inscripció s'inscriu cada Línia caldrà que aquestes en tinguin visibilitat d'atribut. Aquesta visibilitat s'aconsegueix passant la $i:Inscripció$ al mètode `create()` de Línia; i es correspon a l'orientació $Línia \rightarrow Inscripció$ de l'associació s'*inscriu*.

Acoblaments en la recuperació de la Inscripció. El responsable d'enviar el mètode `create()` a la Línia, amb la Inscripció com a argument, és el controlador. Per tant, tant el controlador com la Línia s'acoblen a Inscripció.

Visibilitat sobre la Inscripció. Per tal que el controlador pugui enviar la $i:Inscripció$ activa com argument del missatge `create()` de la Línia cal que en tinguin visibilitat. Això exigeix modificar el model per tal que es compleixi alguna de les següents condicions:

1. El controlador és creador orb d'Inscripció.
2. El controlador pot demanar a la $c:Caminada$ activa que li torni la $i:Inscripció$ activa (per exemple, com valor de retorn de la petició que fa a la $c:Caminada$ de que creï una nova $i:Inscripció$).

⁶Un repositori de Línies fregmentat per Caminades, per exemple, exigeix que les Línies de diferents Inscripcions de la mateixa Caminada no puguin tenir el mateix identificador. És a dir, exigeix un que l'àmbit d'identificació sigui la Caminada. Si el repositori de Línies no està fragmentat per Caminades, l'exigència desapareix. Ara bé, si el repositori de Línies està fragmentat, per exemple, per Inscripcions, usar com a àmbit d'identificació la Caminada complica força les coses. Per tant, la no existència d'un repositori de Línies fragmentat per Caminades no només elimina la necessitat que l'àmbit d'identificació sigui la Caminada, sinó que gairebé exigeix que no ho sigui.

⁷Recordem que l'anàlisi dels *creadors orbs* la presentem per completesa expositiva.

3. La visibilitat d'activació és una visibilitat directa: $GLS \xrightarrow{atr} Inscripció$. Per mantenir aquesta visibilitat cal complir amb alguna de les condicions anteriors.

Recuperació de la Caminada. En el cas d'un repositori centralitzat de Línies, si a més de voler recuperar la Inscripció amb la que que s'inscriu cada Línia, també ens interessa conèixer la Caminada a la que fa referència la $i:Inscripció$ recuperada, caldrà o bé que Inscripció tingui visibilitat de Caminada, o bé que la tingui la pròpia Línia.

Visibilitat d'Inscripció sobre Caminada. Aquí cal analitzar el disseny de l'ES *novaInscripcio(c:Caminada)*. En el disseny que hem desenvolupat fins ara, la Caminada és la creadora de la Inscripció, i per tant tenim visibilitat $Caminada \rightarrow Inscripció$. La visibilitat ara exigida, que és la inversa de la que tenim, es pot aconseguir si quan la Caminada crea la Inscripció es passa a si mateixa com a argument. El resultat és un fort *acoblament mutu* entre Caminada i Inscripció.

Visibilitat de Línia sobre Caminada. Aquesta visibilitat s'aconsegueix passant la Caminada com a argument del mètode *create()* de Línia; i es correspon a l'orientació $Línia \rightarrow Caminada$ d'una associació no existent a MC.

Acoblaments en la recuperació de la Caminada. En el cas d'usar la visibilitat $Inscripció \rightarrow Caminada$, s'introdueix l'acoblament de Inscripció sobre Caminada. En el cas d'usar la visibilitat $Línia \rightarrow Caminada$, el responsable d'enviar el mètode *create()* a la Línia, amb la Caminada com a argument, és el controlador; per tant, tant el controlador com la Línia s'acoblen a Caminada. Ara bé, l'acoblament entre el controlador i Caminada no és nou: en el disseny de l'ES *novaInscripcio(c:Caminada)* ja ha aparegut.⁸

Rèplica del repositori d'Inscripcions. El repositori centralitzat de Línies, cadascuna d'elles amb visibilitat de la Inscripció pertinent, és, en el fons, un *repositori centralitzat d'Inscripcions*. De fet és la *rèplica* del repositori distribuït per Caminades.

Repositori de Caminades. Les solucions proposades per a la recuperació de la Caminada associada a cada Línia converteixen el repositori centralitzat de Línies en un *repositori centralitzat de Caminades* (de fet, en una *rèplica* del repositori de Caminades usat fins el moment). En el cas d'usar la visibilitat $Inscripció \rightarrow Caminada$ s'accedeix a la Caminada a través de la cadena de delegacions $Línia \rightarrow Inscripció \rightarrow Caminada$; en el cas d'usar la visibilitat $Línia \rightarrow Caminada$ s'accedeix a la Caminada directament des de la Línia.

⁸O bé l'argument de l'ES és una realització de Caminada, la qual cosa implica que el controlador té com a mínim visibilitat de paràmetre de Caminada; o bé l'argument de l'ES és un identificador. En aquest darrer cas, el principi de l'*Encaixada* exigeix usar-lo d'immediat per recuperar-ne la Caminada pertinent, i un bon candidat per fer-ho és el controlador.

Superficialitat de l'anàlisi. La necessitat del repositori centralitzat de Línies viola la hipòtesi 106;⁹ l'existència d'un repositori (virtual) centralitzat d'Inscripcions viola la hipòtesi 24.¹⁰ La violació d'una de les hipòtesis de treball deixa sense efecte cap de les decisions preses, i exigeix refer tot el disseny tenint en compte les noves observacions (en concret, la necessitat d'un repositori centralitzat de Línies).

3.4 La Caminada com a creador orb no canònic

Aspectes bàsics del model. Assignar la responsabilitat de creació de la Línia a la Caminada significa acoblar aquesta a Línia. El resultat és un repositori distribuït de Línies on cada Caminada n'és un fragment. Per altra banda s'introdueix una visibilitat d'atribut $\text{Caminada} \xrightarrow{\text{atr}} \text{Línia}$ que no es correspon a la modelització de cap associació de MC.

Recuperació de la Inscripció. En el cas d'un repositori de Línies fragmentat per Caminada, per poder conèixer a quina Inscripció s'inscriu cada Línia caldrà que aquestes en tinguin visibilitat d'atribut. Aquesta visibilitat s'aconsegueix passant la $i : \text{Inscripció}$ al mètode `create()` de Línia; i es correspon a l'orientació $\text{Línia} \rightarrow \text{Inscripció}$ de l'associació s'inscriu.

Acoblaments en la recuperació de la Inscripció. El responsable d'enviar el mètode `create()` a la Línia, amb la Inscripció com a argument, és la Caminada. Per tant, tant la Caminada com la Línia s'acoblen a Inscripció. Però l'acoblament entre Caminada i Inscripció no és nou, ja que Caminada és el *creador* d'Inscripció.

Repositori d'Inscripcions. El repositori de Línies fragmentat en Caminades, on cada Línia té visibilitat de la Inscripció pertinent, és, en el fons, un repositori d'Inscripcions fragmentat per Caminades. El resultat és, de fet, la *rèplica* del repositori distribuït per Caminades que ja teníem, amb la diferència que abans n'hi havia prou en conèixer la Caminada per accedir a les seves Inscripcions, mentre que aquesta rèplica exigeix conèixer també la Línia (el camí d'accés és $\text{Caminada} \rightarrow \text{Línia} \rightarrow \text{Inscripció}$).

Superficialitat de l'anàlisi. La necessitat del repositori de Línies fragmentat per Caminades, viola la hipòtesi 106.¹¹ La violació d'una de les hipòtesis de treball deixa sense efecte cap de les decisions preses, i exigeix refer tot el disseny tenint en compte les noves observacions (en concret, la necessitat d'un repositori de Línies fragmentat per Caminades).

3.5 Identificació en el cas dels creadors orbs

Els principis sobre identificació. Un cop analitzat els dissenys en el cas d'usar un *creador orb* ens queda per analitzar els diferents mecanismes d'identi-

⁹Vegeu pàgina 495.

¹⁰Vegeu pàgines 214.

¹¹Vegeu 495.

ficació en cada cas. El principi de l'*Individualització efectiva* exigeix que la responsabilitat de crear els identificadors (en el cas d'identificació interna) s'assigni al creador de l'objectiu. El principi es basava, entre d'altres, en el fet que les visibilitats exigides eren assegurades amb l'ús d'un *creador canònic*; cosa que ara no és el cas. El principi de la *Consistència individualitzadora* limitava els àmbits d'identificació en aquells que eren consistents amb el MC; la intenció era evitar que per suportar l'àmbit d'identificació s'haguessin d'introduir acoblaments innecessaris.

El controlador: creador i individualitzador sobre Caminada. L'àmbit d'identificació de la Línia que ofereix el controlador com a creadora d'aquesta, és el global, que és incoherent amb els requeriments. Per tal d'usar la Caminada com a àmbit d'identificació caldrà que el controlador delegui en la Caminada pertinent (l'activa en el nostre cas) la responsabilitat de crear l'identificador. I això exigeix un repositori de Línies fragmentat per Caminades; que ha de ser una rèplica del repositori centralitzat exigint pel principi *Creador orb*. La delegació de la responsabilitat d'identificació exigeix que el controlador tingui visibilitat de Caminada; però aquest acoblament ja havia aparegut en el disseny de *novaInscripcio(c:Caminada)*. La necessitat del repositori fragmentat de Línies requereix una visibilitat d'emmagatzematge *Caminada→Línia**, que és del tot nova.

El controlador: creador i individualitzador sobre Inscripció. Si tot i mantenir el controlador com a individualitzador, volem usar la Inscripció com a àmbit d'identificació de la Línia caldrà que el controlador delegui en la Inscripció pertinent (l'activa en el nostre cas) la responsabilitat de crear l'identificador. I això exigeix un repositori de Línies fragmentat per Inscripcions; que ha de ser una rèplica del repositori centralitzat exigint pel principi *Creador orb*. La delegació de la responsabilitat d'identificació exigeix que el controlador tingui visibilitat d'Inscripció; però aquest acoblament ja havia aparegut en el disseny de *novaInscripcio(c:Caminada)*. La necessitat del repositori fragmentat de Línies requereix una visibilitat d'emmagatzematge de *Inscripció→Línia**, que és del tot nova.

La Caminada com a creadora i individualitzadora. L'àmbit d'identificació de la Línia que ofereix la Caminada com a creadora d'aquesta, és coherent amb els requeriments. El principi de la *Consistència individualitzadora* desaconsella l'ús de la Caminada com a àmbit d'identificació perquè augmenta l'acoblament del sistema en introduir un acoblament entre Caminada i Línia. En aquest cas, però, en usar un *creador orb* enlloc d'un *creador canònic*, l'acoblament entre Caminada i Línia ja està present. Per tant, la violació del principi de la *Consistència individualitzadora*, en aquest cas, no té cap efecte negatiu.

3.6 Comparació de les alternatives

Com era d'esperar l'ús del *creador canònic* ens proporciona el model amb menys acoblament. Pel que fa als creadors orbes, l'anàlisi comparativa dels models obtinguts, en termes de nous acoblaments, depèn dels acoblaments ja existents en el nostre disseny.

Responsable	Principi	Observacions
i:Inscripció	<i>Creador canònic</i> <i>Consistència individualitzadora</i>	Repositori de Línies fragmentat per Inscripció Àmbit local al creador; identificació interna
gls:GLS	<i>Creador orb</i>	<ul style="list-style-type: none"> • Repositori de Línies centralitzat • Cal afegir visibilitats per recuperar la Caminada i la Inscripció • Cal afegir visibilitats per a la identificació
gls:GLS:NO	<i>Encarrilament</i> <i>Consistència individualitzadora</i> <i>Baix Acoblament</i>	Cal tenir en compte que alguns dels aco- blaments introduïts potser ja els teníem per altres motius
c:Caminada	<i>Creador orb</i>	<ul style="list-style-type: none"> • Repositori de Línies frag- mentat per Caminades • Cal afegir vi- sibilitats per recuperar la Inscripció • Les visibilitats per la identifica- ció ja han estat afegides per la creació
c:Caminada:NO	<i>Baix Acoblament</i>	Cal tenir en compte que alguns dels aco- blaments introduïts potser ja els teníem per altres motius

Figura 25.1: Creadors de li:Línia¹²
500

4 POST: Enllaç entre i i li ; i activa

Enllaç entre i i li . El *creador* de $li:Línia$ és el seu *creador canònic* $i:Inscripció$; per tant, per definició de creador canònic, també n'és el seu *emmagatzemador canònic*. D'aquí que la creació de $li:Línia$ implica que cal afegir la $li:Línia$ com a destinació del de l'enllaç dirigit $i:Inscripció \rightarrow^* Línia$. Per tant la POST està assegurada.

Manteniment de $i:Inscripció$ activa. El fet que la $i:Inscripció$ estigui activa l'hem modelitzat amb una visibilitat indirecta del controlador a la $i:Inscripció$ des. Mentre no trenquem la cadena de d'enllaços dirigit la $i:Inscripció$ es mantindrà activa.

5 Esdeveniment *fiInscripcio*()

5.1 Contracte *fiInscripcio*()

Contracte <i>fiInscripcio</i> ()
PRE
1. Hi ha una $i:Inscripció$ activa
POST
1. No hi ha cap $Inscripció$ activa

5.2 POST: Cap $Inscripció$ activa

Objectiu a assolir. Per aconseguir la POST n'hi ha prou en trencar la cadena d'enllaços dirigits que va del controlador a la $i:Inscripció$ passant per la $c:Caminada: gls:GLS \rightarrow c:Caminada \rightarrow i:Inscripció$.

Modificació de l'atribut pertinent de $c:Caminada$. La visibilitat de $c:Caminada$ sobre $i:Inscripció$ es pot suprimir fàcilment modificant el valor de l'atribut de $c:Caminada$ que manté la destinació de l'enllaç dirigit $c:Caminada \rightarrow i:Inscripció$.

Notació. Per facilitar l'explicació, anomenarem *InsActiva* a aquest atribut.



Assignació a l'atribut *InsActiva* del valor *null*. Com que la POST exigeix que no hi hagi cap $Inscripció$ activa, hem d'assegurar que l'atribut *InsActiva* de $c:Caminada$ no contingui cap $Inscripció$. Això ho podem fer assignant-li el valor *null*.¹³

¹²Mostrem com a creadors orbs només aquells que no són creadors canònics.

¹³Si el llenguatge de programació no admet els valors *null* assignariem una $Inscripció$ falsa o fictícia.

Caminada activa sense Inscripció activa. Amb l'assignació del valor null a l'atribut `InsActiva` esborrem l'existència d'una `Inscripció` activa. Però encara queda la `Caminada` activa.

Necessitat de desactivar la `c:Caminada`. Cal recordar¹⁴ que l'activació de la `c:Caminada` era una necessitat del disseny per tal de poder modelitzar la petició d'una `Inscripció` activa. Per tant, si ara la `POST` demana desactivar la `i:Inscripció`, també caldrà desactivar la `c:Caminada`.

Desactivació de la `c:Caminada`. Per a desactivar la `c:Caminada` n'hi ha prou en assignar el valor null a l'atribut del controlador que manté la destinació de l'enllaç dirigit `gls:GLS→c:Caminada`.

Necessitat de desactivar la `i:Inscripció`. Cal observar com la desactivació de la `c:Caminada` no és suficient, sinó que també cal desactivar la `i:Inscripció`. Si trenquem l'enllaç dirigit `gls:GLS→c:Caminada` però mantenim l'enllaç dirigit `c:Caminada i:Inscripció` llavors, si en un futur tornem a activar la mateixa `c:Caminada`, automàticament quedarà també activada la `i:Inscripció`!

(120)(119)

Si volem trencar una visibilitat indirecta, i no volem tenir efectes laterals indesitjats, caldrà trencar tots els enllaços dirigits de la cadena

¹⁴Vegeu apartat 13.1. Activació de la `i:Inscripció`, pàgina 235.

Capítol 26

Compleció semàntica

1	^{GLS} Un exemple: la semàntica de <i>fiInscripcio()</i>	506
2	El problema de la compleció semàntica	506
2.1	Estat de compleció	507
2.2	Distinció de repositoris	510
3	Principi de la <i>Incorporació tardana</i>	514
3.1	El principi d' <i>Incorporació tardana</i>	514
3.2	No tothom n'està afectat	515
3.3	Afectació dels contractes	516
3.4	Quan apliquem la <i>Incorporació tardana</i>	518

Contingut detallat del capítol 26

1	<i>GLS</i> Un exemple: la semàntica de <i>fiInscripcio()</i>	506
	Responsabilitat del CU	506
	Visió de l'usuari i visió de l'anàlisi	506
	La visió de l'usuari	506
	Responsabilitat de l'ES <i>fiInscripcio()</i>	506
	Desajust entre la visió de l'usuari i el disseny	506
	Convergència de les dues visions dels ES	506
2	El problema de la compleció semàntica	506
	Plantejament del problema	506
2.1	Estat de compleció	507
2.1.1	Ús d'un atribut com a solució a la compleció semàntica	507
	Estat de compleció	507
	Atribut de compleció amb valor lògic	507
	Atribut de compleció amb valors predefinits	507
2.1.2	Canvis en els contractes	508
	Afectació dels contractes	508
	Oblit de l'especificació	508
	Control a les PRE	508
	Finalització en les POST	508
	Selecció d'estat en les PRE	508
2.1.3	<i>GLS</i> Canvis en els contractes del CU <i>ferInscripció</i>	508
	Excursió. (Múltiples estats de compleció)	509
2.1.4	Mantenibilitat en el cas d'usar l'estat de compleció	510
	Simplicitat barroera	510
	Excursió. (Autòmat d'estats)	510
2.2	Distinció de repositoris	510
2.2.1	Estat de compleció <i>no finalitzada</i> local a un sol CU	510
	Consideració simplificada	510
	CU de creació	510
	CU de creació i estat <i>finalitzat</i>	510
	POST de finalització	510
	Localitat de l'estat de compleció <i>no finalitzat</i>	511
	Estat de compleció supervivent	511
	Cicle de vida dels estats de compleció	511
	Excursió. (Localitat de la no finalització)	511
2.2.2	Repositoris heterogenis i repositoris homogenis	512
	Barreja d'objectes amb diferent <i>estat de compleció</i>	512
	Consulta contínua a l'estat de compleció	512
	Problemàtica de l'especificació i del disseny	512
	Repositoris segons l' <i>estat de compleció</i>	512
2.2.3	Incorporació de nous objectes al sistema	512
	Ús de repositoris homogenis segons l'estat de compleció	512
	Responsabilitat d'incorporar un nou objecte al sistema	513

	Responsabilitat de crear els enllaços incidents	513
	Alternativa a l'atribut de compleció	513
	Incorporació tardana	513
	Separació de repositoris i incorporació tardana	514
3	Principi de la <i>Incorporació tardana</i>	514
	El que segueix	514
3.1	El principi d' <i>Incorporació tardana</i>	514
	Incorporació tardana	514
	Enllaços afectats	514
	Quan s'exerceix la responsabilitat	514
	<i>Incorporació tardana</i> i distinció de repositoris	515
	Extensibilitat	515
3.2	No tothom n'està afectat	515
	Una formulació estranya	515
	El perquè dels enllaços incidents	515
	Creació d'una estructura	515
	Lligams incidents no problemàtics	515
	Enllaços dirigir afectats per la <i>Incorporació tardana</i>	516
	El perquè dels enllaços dirigits d'atribut	516
3.3	Afectació dels contractes	516
	Posposem l'emmagatzematge	516
	El cas del creador canònic	517
	Ultrapassem les fronteres del disseny	517
	El principi de l' <i>Estabilitat de l'especificació</i>	517
	Un principi que surt reforçat	517
3.4	Quan apliquem la <i>Incorporació tardana</i>	518
	Anàlisi prèvia al disseny	518
	Altres afectacions	518
	Una màxima a tenir present	518
	Ordenació de prioritats	518

1 ^{GLS}Un exemple: la semàntica de *fiInscripcio()*

Responsabilitat del CU. El CU *ferInscripció* té la responsabilitat d'introduir al sistema una nova *Inscripció* i tota la seva informació associada; en concret, les *Línies* que s'inscriuen en la *c:Caminada* en virtut d'aquesta *Inscripció*.

Visió de l'usuari i visió de l'anàlisi. L'anàlisi del problema ens ha portat a identificar els conceptes *Inscripció*, *Caminada* i *Línia d'inscripció*. Per l'usuari, però, la seva idea d'*Inscripció* és molt més global que el concepte *Inscripció* que apareix en l'anàlisi. En concret, la *Inscripció* engloba les *Línies* i la *Caminada* a la que es refereix la *Inscripció*.¹

La visió de l'usuari. Així, des del punt de vista de l'usuari la *Inscripció* no s'ha introduït del tot al sistema fins que no s'han introduït totes les *Línies* corresponents, la qual cosa és coherent amb la responsabilitat explicitada del CU *ferInscripció*.

Responsabilitat de l'ES *fiInscripcio()*. L'ES *fiInscripcio()* és el mecanisme de què disposa l'usuari per indicar que ja ha acabat d'introduir les dades corresponents a la nova *Inscripció*.

Desajust entre la visió de l'usuari i el disseny. En el disseny hem introduït els components *Inscripció*, *Caminada* i *Línia* com a models software dels conceptes del mateix nom. En aquest sentit la creació d'una nova realització d'*Inscripció*, i la seva incorporació al sistema via emmagatzematge, pot donar a l'usuari la falsa idea que en el sistema s'ha introduït una nova realització de la seva idea d'*Inscripció*. Des del punt de vista de la correspondència entre especificació i disseny, efectivament s'ha introduït al sistema una nova realització d'*Inscripció*; però des del punt de vista de l'usuari, només hem introduït *parcialment* la seva idea d'*Inscripció*: encara manquen, com a mínim, les *Línies*.

Convergència de les dues visions dels ES. L'ES *fiInscripcio()* és el mecanisme que l'usuari usa per indicar que per ell la *Inscripció* està complerta. En el disseny caldrà utilitzar aquesta informació per fer coincidir les visions de l'usuari i el disseny. Per exemple, el disseny haurà d'impedir que es puguin fer peticions sobre les *Inscripcions* que no hagin rebut, encara, el missatge *fiInscripcio()*.

2 El problema de la compleció semàntica

Plantejament del problema. L'anàlisi desglossa en diferents *conceptes*, i el disseny en els corresponents *components*, el que per l'usuari és una *idea simple*.²

¹De fet l'usuari pot ser que descongui el concepte *Línia* i que en el seu lloc usi el concepte *Participant*.

²L'especificació descriu el problema. Però aquesta *descripció* no deixa de ser un *model*. Així l'especificació és un model descriptiu del problema; i el *disseny* és un *model col·laboratiu* sobre la

La introducció en el sistema del que per l'usuari és una unitat, pot convertir-se en el disseny en la creació de diferents realitzacions. La consistència entre el sistema i la visió de l'usuari exigeix identificar o marcar aquelles realitzacions complertes pel que fa al sistema, però incomplertes pel que fa a l'usuari.

Com podem indicar que una realització d'una abstracció³ es correspon a una idea completa des del punt de vista de l'usuari?

(107) ?

2.1 Estat de compleció

2.1.1 Ús d'un atribut com a solució a la compleció semàntica

Estat de compleció. Una possible solució per indicar que una realització està completa des del punt de vista de l'usuari és la d'usar un atribut d'aquesta realització que ens indiqui l'*estat de compleció* de l'objecte en qüestió.

Definició. Estat de compleció

L'estat de compleció és el valor associat a un objecte que indica fins a quin punt es correspon a una realització completa des del punt de l'usuari.

Definició. Atribut de compleció

Anomenem atribut de compleció aquell atribut que manté l'estat de compleció.

Atribut de compleció amb valor lògic.⁴ En el cas més simple l'*atribut de compleció* només té dos valors: *cert* i *fals*. El valor *cert* indica que l'objecte està acabat des del punt de vista de l'usuari; el valor *fals* indica que la realització està inacabada des del punt de vista de l'usuari. D'alguna manera l'ús d'un *atribut amb valor lògic* ens indica si la realització es troba o no en un estat determinat; en concret en l'estat de realització acabada des del punt de vista de l'usuari.

Atribut de compleció amb valors predefinits. Sovint la compleció d'una realització passa per diferents estats de compleció parcial. En aquests casos l'*atribut de compleció* contindrà el valor de l'estat en qüestió.

descripció del problema. En el nostre exemple, *Inscripció* és un element del problema; *Inscripció* n'és el model que empren per descriure el problema; i *Inscripció* és el model, en termes de la solució, de *Inscripció*.

³En l'especificació haurem d'indicar que una entitat és completa des del punt de vista de l'usuari; en el disseny haurem d'indicar que un objecte és complet des del punt de vista de l'usuari.

⁴Un atribut de valor lògic és aquell que només pot tenir dos valors diferents, generalment representats per *cert* i *fals*.

Exemple 107 (Estat de compleció) *En el cas de la Inscripció, l'atribut de compleció pot tenir els valors següents: sense participants, sense responsable (però amb participants), i finalitzada.*

2.1.2 Canvis en els contractes

Afectació dels contractes. Sovint l'ús de l'estat de compleció és una decisió de l'especificació: en el moment d'analitzar el problema veiem la discrepància entre els *conceptes* que proposem i les *idees* de l'usuari, i prenem les precaucions necessàries per tractar correctament aquesta discrepància. En aquest cas els contractes ho han de contemplar.

Oblit de l'especificació. Pot ser que l'estat de compleció aparegui com una necessitat del disseny sense que ens l'haguéssim plantejat en l'especificació. Llavors → caldria analitzar si ha estat un oblit de l'especificació, o es tracta d'unes exigències que només tenen sentit en l'àmbit del disseny. Però l'estat de compleció està definit en termes de l'usuari i del problema, per tant si el primer com que ens apareix és en el disseny, llavors és clarament un oblit de l'especificació. En conseqüència el principi de la *Estabilitat de l'especificació* → exigeix aplicar la reconsideració als contractes de l'especificació.

12.2.Manteniment consistent dels contractes,
pàgina 224

Principi, pàgina 227

(121)

L'estat de compleció, en cas d'usar-lo, ha d'aparèixer en els contractes

Control a les PRE. Les operacions que ajuden a completar, des del punt de vista de l'usuari, una realització, han de contenir en la PRE l'exigència que la realització en qüestió *no estigui* en l'estat *finalitzat*.

Finalització en les POST. Les operacions que, des del punt de vista de l'usuari, finalitzen la realització, han de contenir una POST que assegurí que la realització *està* en l'estat *finalitzat*.

Selecció d'estat en les PRE. A més, tota operació sobre la realització ha de tenir com a PRE l'exigència que la realització estigui en l'estat *finalitzat*. Només en el cas d'operacions de compleció, o de naturalesa molt especial (per exemple, la destrucció d'una realització no acabada del tot) les PRE admetran casos amb estat de compleció diferent de *finalitzat*.

2.1.3 ^{GLS}Canvis en els contractes del CU *ferInscripció*

Tot seguit mostrem els contractes del CU *ferInscripció* amb els canvis acabats d'esbossar. Considerem el cas d'un sol estat de compleció (atribut amb valor lògic).⁵

⁵Aquests contractes tant són de l'especificació com del disseny. A 2.1.El punt de partida del disseny, pàgina 22, → hi ha una explicació de la diferència entre els contractes de l'especificació i els contractes del disseny.

Contracte ***novaInscripció*** (*c: Caminada*)

Paràmetres

- *c*: Caminada a la que es vol fer la inscripció

PRE

1. No hi ha cap *Inscripció* activa

POST

1. Hi ha una nova realització *i: Inscripció*
2. Hi ha un enllaç entre *i* i *c*, corresponent a l'associació relativa a
3. La inscripció *i* està activa
4. L'estat de compleció de *i* és *no finalitzat*

Contracte ***nouParticipant*** (*nom, edat*)

Paràmetres

- *nom*: Nom del participant
- *edat*: Edat del participant

PRE

1. Hi ha una *i: Inscripció* activa
2. L'estat de compleció de *i* és *no finalitzat*

POST

1. Hi ha una nova realització *li: Línia*
2. Hi ha un enllaç entre *i* i *li*, corresponent a l'associació *inscrit* en
3. La inscripció *i* està activa
4. L'estat de compleció de *i* és *no finalitzat*

Contracte ***fiInscripció*** ()

PRE

1. Hi ha una *i: Inscripció* activa
2. L'estat de compleció de *i* és *no finalitzat*

POST

1. No hi ha cap *Inscripció* activa
2. L'estat de compleció de *i* és *finalitzat*

Excursió. (Múltiples estats de compleció) L'ús de diferents estats de compleció permet donar més robustesa al sistema. Per exemple permetria fàcilment evitar la compleció d'una *Inscripció* (us de l'ES *novaInscripció* (*c: Caminada*)) en el cas de no haver-hi introduït cap *Participant*.

2.1.4 Mantenibilitat en el cas d'usar l'estat de compleció

Simplicitat barroera. *L'estat de compleció* és una solució simple. Però “embruta” l'especificació i el disseny: en tota operació sobre una realització cal analitzar quins valors de l'estat de compleció requereix o admet, i en quin nou estat de compleció la deixa. Com a conseqüència la mantenibilitat se'n ressent.

Excursió. (Autòmat d'estats) Una manera alternativa de tractar els diferents estats de compleció és mitjançant un *autòmat finit*. En els contractes de les operacions no apareix per a res l'estat de compleció; i de manera independent als contractes es dissenya un *autòmat finit* on cada node és un estat de compleció, i els arcs són els diferents ES.

Llavors, les condicions de correcció per a l'ús d'un ES exigeixen el compliment del es PRE i el fet d'estar en un estat de l'autòmat amb transició definida per l'ES en qüestió; l'execució d'un ES implica la realització de la transició corresponent dins de l'autòmat d'estats.

Aquest mecanisme, en desacoblar la semàntica dels contractes dels ES del control i de la gestió de la compleció des del punt de vista de l'usuari, assegura la mantenibilitat.

El mecanisme de l'*autòmat d'estats* també es pot usar per assegurar la *robustesa de la seqüenciació*. En desacoblar la semàntica dels contractes dels ES del control de la seqüència del CU, donem molta més flexibilitat als ES, facilitem la mantenibilitat, i oferim una solució més simple i clara.

2.2 Distinció de repositoris

2.2.1 Estat de compleció no finalitzada local a un sol CU

Consideració simplificada. Sense pèrdua de generalitat, a partir d'ara considerem un estat de compleció lògic, i anomenarem als seus valors *finalitzat* i *no finalitzat*. En el cas que l'estat de compleció no sigui lògic (i per tant tingui més de dos valors) sempre podem definir una bipartició entre els valors de l'estat de compleció que es corresponguin a un objecte no complert, i els valors de l'estat de compleció que es corresponguin a un objecte complert.

CU de creació. Sovint la responsabilitat d'un CU és la creació complerta, des de la visió de l'usuari, d'una realització. En aquests casos (que anomenem *CU de creació*) els diferents ES col·laboren per crear totes les realitzacions i lligams necessaris per obtenir el que seria una realització complerta segons la visió de l'usuari.⁶

CU de creació i estat finalitzat. Si la responsabilitat del CU és la creació complerta, des de la visió de l'usuari, d'una realització, podem assumir que un cop aconseguit l'objectiu no tornem enrera. És a dir, dins d'un CU de creació un cop s'arriba a l'estat *finalitzat* aquest es manté fins a acabar el CU. En cas contrari significa que l'usuari pot reconsiderar la finalització; per tant, en realitat, l'estat de compleció correcte seria *aparentment finalitzat*, que podríem encabir dins dels estats corresponents a objectes no complerts.

⁶En especificació els ES es preocupen d'assegurar l'existència de les entitats i enllaços pertinents. En el disseny els ES es preocupen de crear els objectes i enllaços dirigits que pertoquein.

POST de finalització. En un *CU de creació* la responsabilitat del CU exigeix deixar la nova realització en l'estat de compleció *finalitzat*. De fet aquesta ha de ser una de les POST de l'ES que l'usuari usa per indicar que dóna per acabada la creació. L'anomenarem *POST de finalització*.

Localitat de l'estat de compleció *no finalitzat*. El *CU de creació* crea i finalitza una realització. Per tant, tots els estats de compleció *no finalitzat* possibles d'aquesta realització⁷ s'han de trobar localitzats dins la realització del CU en qüestió.

Estat de compleció supervivent. La responsabilitat primera d'un *CU de creació* és la creació completa d'una nova realització; per tant l'estat *finalitzat* "sobreviu" al CU. A més, la localitat de l'estat de compleció *no finalitzat* assegura que l'estat *finalitzat* és l'únic estat de compleció que "sobreviu" al CU.

Cicle de vida dels estats de compleció. L'observació de la localitat de l'estat de compleció *no finalitzat* i de la supervivència de l'estat de compleció *finalitzat* en el cas d'un *CU de creació* es pot generalitzar en els termes següents:

Cicle de vida de l'estat de compleció.

Donada una realització qualsevol d'un component, els estats de compleció associats a la idea de *no finalitzat* queden localitzats dins la realització d'un CU. Només els estats de compleció associats a la idea de *finalitzat* poden ser compartits per diferents CU.

(122)

Excursió. (Localitat de la *no finalització*) En els *CU de creació* ho acabem de veure. En un CU de destrucció de la realització l'afirmació és prou evident. En el cas d'un CU que tingui com a objectiu modificar una realització pot ser que la modificació exigeixi com a etapa temporal la destrucció de lligams abans de crear els nous lligams, per exemple. Sigui com sigui durant la modificació la realització pot estar temporalment en un estat *no finalitzat*, però un cop s'hagi acabat la modificació la realització recuperarà l'estat *finalitzat*.

Un cas poc habitual és que la modificació es faci amb més d'un CU. Llavors podríem tenir que un CU deixa la realització en un estat *no finalitzat* i que un altre CU el finalitza. Es tracta d'una situació poc habitual i que per tant no invalida la idea bàsica de la localitat de la *no finalització*.

En cas de trobar-nos davant un d'aquests estranyes situacions en les que no es compleix la localitat de la *no compleció*, sempre podem considerar dos tipus d'estat finalitzat: l'*estat finalitzat* i l'*estat finalitzat parcialment*, on aquest darrer correspon al d'un objecte finalitzat segons un CU, però no finalitzat del tot segons l'usuari. D'aquesta manera mantenim les propietats desitjades.

⁷Més concretament, tots els moments en què la realització es correspon a la idea d'una realització no finalitzada, i per tant en els que el valor de l'estat de compleció de la realització és *no finalitzat*.

2.2.2 Repositoris heterogenis i repositoris homogenis

! (108)

Fins aquí tant estàvem parlant d'especificació com de disseny. En aquest apartat, que parlem de repositoris i per tant d'emmagatzematge, està clar que estem parlant de *disseny*.

Barreja d'objectes amb diferent estat de compleció. L'ús de l'*estat de compleció* significa que els repositoris emmagatzemen conjuntament tant objectes *finalitzats* com objectes *per finalitzar*; es tracta de *repositoris heterogenis*.

Consulta contínua a l'estat de compleció. Com que els objectes *finalitzats* i els objectes *per finalitzar* només es distingeixen pel valor de l'atribut que manté l'*estat de compleció*, caldrà preguntar pel valor d'aquest atribut en el moment de fer qualsevol petició sobre l'objecte.

Problemàtica de l'especificació i del disseny. Com a problemàtica de l'especificació, la gestió dels estats de compleció exigeix un mecanisme de control com l'aparició de condicions específiques en els contractes. En el disseny, en el moment d'accedir a qualsevol realització caldrà consultar el seu estat de compleció per determinar quines operacions li són aplicables.

Repositoris segons l'estat de compleció. La solució per evitar estar preguntant contínuament l'estat de compleció passa per usar repositoris diferents per cada estat diferent de compleció; és a dir, *repositoris homogenis*. És important observar com els repositoris d'objectes *finalitzats* són aquells que contenen justament els objectes sobre els que ha de treballar el sistema.⁸

2.2.3 Incorporació de nous objectes al sistema

! (109)

En aquest apartat parlem d'objectes i repositoris. Per tant, estem parlant de *disseny*.

Ús de repositoris homogenis segons l'estat de compleció. La idea de tenir repositoris diferents pels objectes *finalitzats* i pels objectes *no finalitzats*, per mor de la localitat de la no finalització, es pot expressar en els termes següents:

(123)

Ús de repositoris segons l'estat de compleció.

Cada CU és responsable de mantenir, en repositoris locals al CU, els objectes amb estat de compleció *no finalitzat* que li són propis; i d'in-

⁸Els objectes *per finalitzar* són objectes en un estat transitori, que esperen a què l'usuari els *finalitzi*.

corporar en repositoris del sistema els objectes que atenyin l'estat de compleció *finalitzat*.

Responsabilitat d'incorporar un nou objecte al sistema. En un CU de creació sabem segur que l'objecte creat acaba tenint l'estat de compleció *finalitzat*. Per tant és responsabilitat del CU de creació, i més concretament de l'ES que conté la POST de finalització de l'objecte, la incorporació de l'objecte al sistema.

Definició. Enllaços incidents i sortints

Sigui una associació assoc entre A i B, modelitzada amb una visibilitat d'atribut de $A \xrightarrow{atr} B$. Llavors direm que l'enllaç dirigit $a : A \rightarrow b : B$, que és una realització de la visibilitat $A \rightarrow B$, és un enllaç dirigit sortint de $a : A$ i incident a $b : B$.

Responsabilitat de crear els enllaços incidents. La incorporació al sistema significa emmagatzematge, i per tant visibilitat d'atribut des del repositori pertinent. Així, és responsabilitat de l'ES que conté la POST de finalització la creació dels *enllaços incidents* a l'objecte creat pel CU de creació. Abans no s'havien pogut crear els enllaços incidents perquè l'estat de compleció era *no finalitzat*, i els repositoris del sistema només contenen objectes *finalitzats*.

Alternativa a l'atribut de compleció. Totes aquestes observacions permeten definir un mecanisme alternatiu al manteniment via atribut de l'estat de compleció, basat en la localitat de la no finalització. La idea és esperar que l'usuari indiqui, amb l'ES pertinent, que la creació ha finalitzat, per incorporar la nova realització al sistema. És a dir, els diferents ES del CU van construint el nou objecte i generen tots els enllaços sortints; però els enllaços incidents (com els de l'emmagatzematge) no es creen fins que l'usuari no indiqui la finalització de la creació.

Alternativa a l'estat de compleció.

- *Separació de repositoris.* En el CU de creació mantenim els objectes a mig crear en repositoris locals al CU
- *Incorporació tardana.* En l'ES que conté la POST de finalització de l'objecte és el moment en què els nous objectes es treuen del repositori local al CU i es posen en el repositori corresponent d'objectes finalitzats

(124)

Incorporació tardana. La *incorporació tardana* acabada d'exposar és el resultat de l'aplicació del principi de la *Incorporació tardana*, i que presentem en el

següent apartat. Aquí només hem analitzat el cas del repositori d'emmagatzematge, però

Separació de repositoris i incorporació tardana. L'aplicació del principi de l'*Aplicació tardana* generalment exigeix crear l'objecte en un moment, i incorporar-lo en el sistema, és a dir, afegir-lo en el repositori d'objectes creats, en un altre moment. I entre un moment i l'altre què passa? Doncs que l'objecte incomplet creat l'hem de mantenir amb visibilitat d'atribut. És a dir, l'*Incorporació tardana* implica, necessàriament, un emmagatzematge local al CU de creació; i per tant la *separació de repositoris*.

(125)

Incorporació tardana i separació de repositoris.

L'aplicació del principi de la *Incorporació tardana* implica que mantenim les realitzacions inacabades en *repositoris separats* de les realitzacions finalitzades.

Incorporació tardana \implies Separació de repositoris

3 Principi de la *Incorporació tardana*

El que segueix. En el aquest apartat presentem el principi de la *Incorporació tardana*, que es pot veure com una generalització de les necessitats esbossades per diferenciar els objectes complets dels objectes per completar sense usar l'estat de compleció.[→] Presentarem el principi, veurem com resol la qüestió de la compleció, i finalment analitzarem en deteniment la seva formulació.

26.2.2. Distinció de repositoris, pàgina 510

3.1 El principi d'*Incorporació tardana*

Principi



Incorporació tardana. La creació dels enllaços d'atribut incidents⁹ des d'un objecte preexistent a un objecte nou l'endarrerim tant com sigui possible

Enllaços afectats. La *Incoirporació tardana* afecta tot enllaç dirigit amb el següent aspecte:

$\text{antic} \xrightarrow{\text{atr}} \text{nou}^*$

Definició pàgina 513

⁹La definició d'*enllaç incident* \rightarrow s'ha fet en termes de visibilitats d'atribut. Per tant aquí no caldria demanar que l'enllaç dirigit sigui d'atribut. De totes maneres en explicitar-ho evitem problemes d'interpretació del principi.

Quan s'exerceix la responsabilitat. Cal observar que aquest principi no dóna propostes d'assignació de responsabilitat sobre els objectes o components. No ens dóna propostes sobre *qui* ha d'exercir determinada responsabilitat, sinó que posa restriccions a *quan* s'ha d'exercir.

Incorporació tardana i distinció de repositoris. La solució al problema de la compleció semàntica basada en l'ús de repositoris homogenis segons l'estat de compleció, i que hem sumaryt a la conclusió 124, \rightarrow es pot veure com una generalització del principi de la *Incorporació tardana*: allà només parlàvem dels enllaços d'emmagatzematge; aquí parlem dels enllaços incidents (que engloben els d'emmagatzematge).

Conclusió 124, pàgina 513. (26.2.2. Distinció de repositoris)

Extensibilitat. El principi de la *Incorporació tardana* facilita l'extensibilitat dels CU de creació. Per exemple, afegir un ES que cancel·li tota la creació feta fins el moment és força simple: com que no s'ha creat cap enllaç incident des dels repositoris del sistema, només cal netejar les estructures locals del CU.

3.2 No tothom n'està afectat

Una formulació estranya. En la formulació del principi de la *Incorporació tardana* parlem dels enllaços incidents des d'un objecte preexistent a un objecte nou. Perquè?

El perquè dels enllaços incidents. El fet de posposar els enllaços incidents a un nou objecte resol dues qüestions:

- *Repositoris homogenis.* Només afegim en el repositori d'emmagatzematge quan l'objecte està complert des del punt de vista de l'usuari
- *Separació i localitat.* Els accessos als objectes incomplets només són possibles des de la localitat del CU de creació. Mai un procés pot treballar amb objectes complets i incomplets simultàniament.

Creació d'una estructura. Imaginem el cas on la idea de completesa per part de l'usuari coincideix amb una estructura complexa, tota ella lligada a un sol objecte.

Exemple 108 (Creació d'una estructura) Una *Inscripció* és, internament, una estructura formada per una *Inscripció* i múltiples *Partíceps*. Tots els diferents *Partíceps* estan lligats a una sola $i : \text{Inscripció}$, la que està creant el CU. Aquest lligam es concreta en l'emmagatzematge $i : \text{Inscripció} \rightarrow \text{Partícep}^*$.

Què passa amb aquest emmagatzematge? Cal posposar-lo també?

Lligams incidents no problemàtics. En el cas d'una estructura nova amb una única "arrel" els enllaços dirigits sortints d'aquesta arrel no són mai accessibles des d'un repositori d'objectes amb estat de finalització *finalitzat*. Això és així perquè l'únic camí per arribar a aquests enllaços dirigits és a través de l'arrel única de l'estructura, arrel que no està en cap repositori d'objectes finalitzats.

Exemple 109 (Enllaços incidents no problemàtics) En el CU *ferInscripció* un ES crea la $i:Inscripció$ però no l'emmagatzema en el repositori d'Inscripcions. Això significa que la $i:Inscripció$ només és accessible dins l'emmagatzematge temporal que manté el CU de creació.

Un altre ES crea un $p:Partícep$ i l'afegeix al repositori de *Partíceps* de i . És a dir, afegeix p a l'enllaç dirigit $i:Inscripció \rightarrow Partícep^*$.

Fixem-nos però que tot i haver creat un enllaç incident a un objecte nou (l'enllaç dirigit $i:Inscripció \rightarrow Partícep^*$ sobre $p:Partícep$) aquest nou objecte només és accessible a partir de $i:Inscripció$ que també és un objecte nou. Però $i:Inscripció$ no és accessible des de cap dels repositoris d'objectes complets.

Anem iterant aquest ES. Al final tenim un enllaç dirigit $i:Inscripció \rightarrow Partícep^*$ que abasta múltiples *Partíceps*. I l'única manera d'accedir-hi és des de la $i:Inscripció$, que al seu torn només és accessible des de la localitat del CU de creació.

Enllaços dirigir afectats per la Incorporació tardana. Les observacions anteriors justifiquen que només ens proposem de posposar la creació dels enllaços dirigits que surtin d'un objecte preexistent al CU de creació i incideixin en un dels objectes nous creats per aquest CU.

El perquè dels enllaços dirigits d'atribut. Els únics enllaços dirigits que ens preocupen són els d'atribut perquè són els que transcendeixen el context d'emissió. I és que les noves realitzacions que creem les creem per usar-les en un futur. La creació d' $ús^{\rightarrow}$ és un recurs del disseny per resoldre una qüestió local a l'ES, i per tant no té sentit parlar d'objecte complet o incomplet: l'objecte un cop acabat el matem!.

3.3 Afectació dels contractes

Posposem l'emmagatzematge. El principi de la *Incorporació tardana* demana que uns determinats enllaços dirigits (els incidents al nou objecte) s'exigeixin com més tard millor. Aquests enllaços dirigits són, com a mínim, els d'emmagatzematge. Per tant, la *Incorporació tardana* demana posposar, com a mínim, l'emmagatzematge.

Exemple 110 (Enllaços incidents) *Suposem que volem tenir constància de l'albarà de més import generat. Llavors cada cop que creem un nou albarà, a part d'emmagatzemar-lo, caldrà comparar el seu import amb l'albarà que fins ara era el d'import superior. El resultat és que el CU de creació genera sempre un enllaç dirigit incident al nou albarà (l'enllaç d'emmagatzematge) i a vegades un segon enllaç incident (el que indica que de moment aquest és l'albarà de major import).*

El cas del creador canònic. Si estem usant el *Creador canònic* llavors tot enllaç d'emmagatzematge és el model d'un enllaç. I per tant segurament en l'especificació hi haurà una POST que demana l'existència de l'enllaç entre l'emmagatzemador i l'objecte emmagatzemat. Però com que estem especificant no podem saber qui és emmagatzemador i qui objecte emmagatzemat.

Exemple 111 (Incorporació tardana i reconsideració) *El creador canònic de la Inscripció és la Caminada; per tant hi ha una visibilitat d'emmagatzematge Caminada→Inscripció*.*

La visibilitat d'emmagatzematge és el model de l'associació referent: Caminada-Inscripció.

En l'especificació, en el CU de creació de la i:Inscripció, hi ha una POST que demana que la nova i:Inscripció estigui enllaçada a la c:Caminada indicada. Formalment, referent(c,i).

En l'especificació exigim l'enllaç; en el disseny aquesta exigència coincideix amb les necessitats d'emmagatzematge, i es concreta en la presència d'un enllaç dirigit $c : C \rightarrow I^$ tal que $i : I$ forma part de la destinació de l'enllaç dirigit.*

L'alicació de la Incorporació tardana en el disseny significa que la POST de l'especificació que demana lligar c i i es posposi al darrer ES, que és qui conté la POST de finalització.

Ultrapassem les fronteres del disseny. Les observacions anteriors porten a dir que l'ús del principi de la *Incorporació tardana* juntament amb el principi del *Creador canònic* exigeixen reconsiderar l'especificació. En concret, la POST de finalització de la nova realització que apareix en l'especificació, cal exigir-la en el moment més tardà possible.

El principi de l'Estabilitat de l'especificació. El principi de l'Estabilitat de l'especificació[→] diu que les exigències del disseny, si no són oblidats de l'especificació, s'introdueixen en forma d'anotació sobre els contractes. En virtut d'aquest principi caldrà afegir una nota que digui que la visibilitat d'emmagatzematge corresponent al creador canònic es veu afectada per la *Incorporació tardana*.

Un principi que surt reforçat. I si el creador és un creador orb? La visibilitat d'emmagatzematge també es veu afectada per la *Incorporació tardana*

però ara intentar traspasar aquesta exigència a l'especificació es fa com bastant complex: caldria demanar a l'especificació l'existència d'enllaços que no es corresponen a cap de les associacions de *MC*!!¹⁰ El principi de l'*Estabilitat de l'especificació* que ens diu que no hem de modificar l'especificació per expressar la incorporació tardana, davant de casos com aquest pren molta força.

3.4 Quan apliquem la *Incorporació tardana*

Anàlisi prèvia al disseny. Abans de construir tot el disseny, només mirant l'especificació podem saber que caldrà cercar un creador i caldrà mantenir actiu l'objecte creat. Amb aquesta informació en tenim prou per, sense construir cap diagrama de comunicació, detectar enllaços dirigits que quedaran afectats per la *Incorporació tardana*.

Altres afectacions. Però potser en el disseny es generen altres enllaços dirigits, que a priori no acabem de veure. I potser alguns d'aquests enllaços dirigits també estan afectats per la *Incorporació tardana*. Per tant hem de construir, primer, el disseny.

(126)

Aplicació tardana de la *incorporació tardana*.

- L'experiència pot permetre detectar, abans o durant la construcció del disseny, on hi ha afectació per la *Incorporació tardana*
- Molt més segur, però, és primer construir el disseny, analitzar els enllaços dirigits generats, i tot seguit modificar el disseny tot aplicant la *Incorporació tardana*

Una màxima a tenir present. La proposta d'aplicar tardanament la *Incorporació tardana* és consistent amb una de les premisses bàsiques de tot desenvolupament:

Primer el que toca; després el que toca després

Ordenació de prioritats. Aquesta màxima diu: primer aconseguir que el disseny faci el que pertoca; després ja el modificaràs per tal que, per exemple:

- Sigui més eficient. (Principi de l'*Eficiència en segon pla*[→])
- La compleció dels objectes sigui coherent amb la compleció semàntica. (Principi de la *Incorporació tardana*[→])
- Millori l'acoblament o la cohesió. (Principi de la *Fabricació pura*[→])

12.3.2. Perill de l'eficiència prematura, pàgina 228

26.3. Principi de la *Incorporació tardana*, pàgina 514

??, pàgina ??

¹⁰Recordem que estem considerant un creador orb, i que per tant la visibilitat d'emmagatzematge viola *Espill*.

Capítol 27

GLS El CU ferInscripció amb *Incorporació tardana*

1	Els contractes fins al moment	522
2	Anàlisi de la <i>Incorporació tardana</i>	523
3	Un disseny amb <i>Incorporació tardana</i>	524
3.1	Decisions prèvies	524
3.2	Anàlisi dels enllaços dirigits d'atribut	525
3.3	Canvis en el disseny	526
3.4	Afectació en els contractes contractes	528

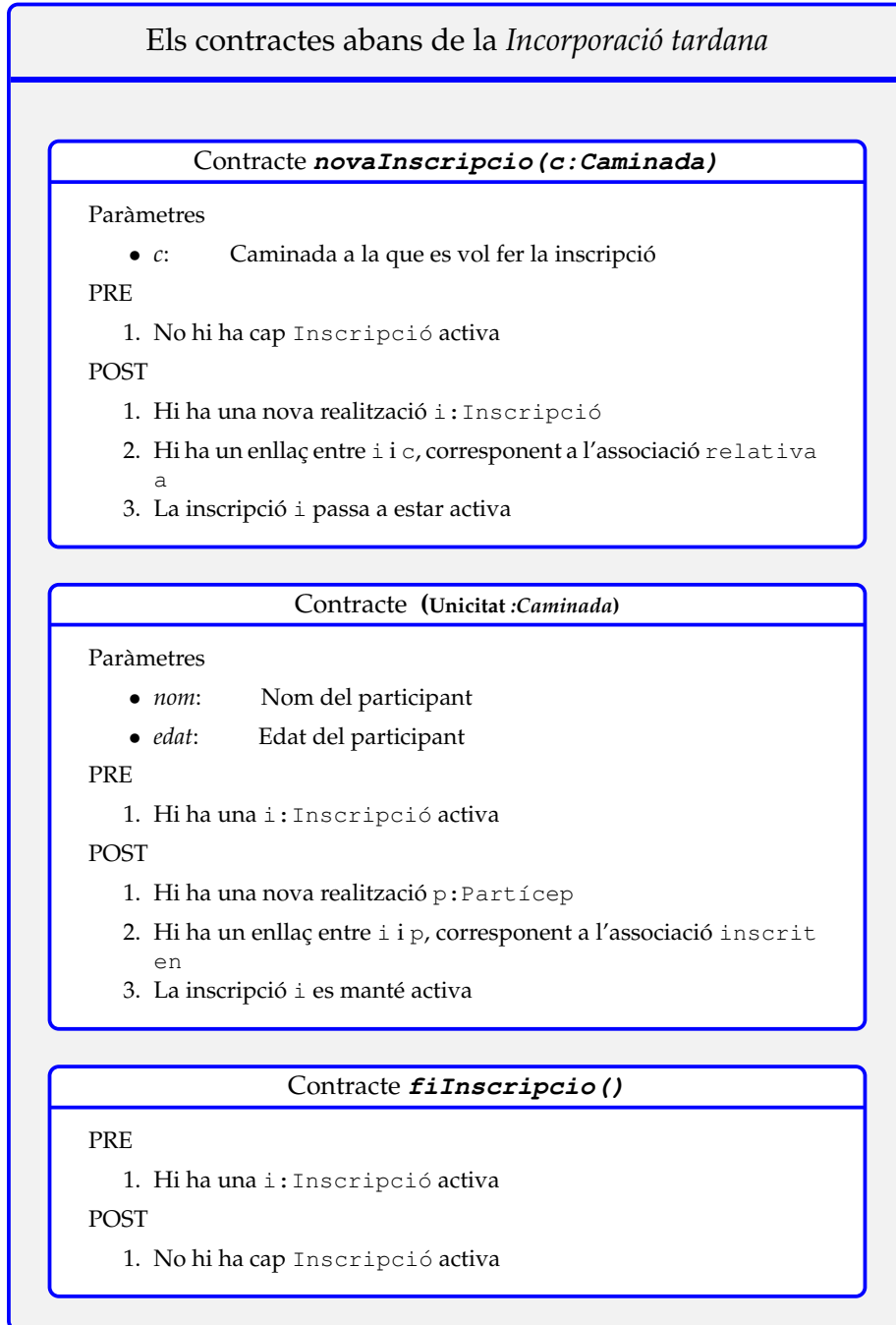
Contingut detallat del capítol 27

1	Els contractes fins al moment	522
2	Anàlisi de la <i>Incorporació tardana</i>	523
	Noves realitzacions	523
	Enllaços dirigits generats pel disseny	523
	Anàlisi de creadors	523
	Enllaços dirigits que cal analitzar	523
	Enllaços dirigits exigits per l'especificació	523
	Una resposta clara	523
	Aplicació al nostre cas	523
3	Un disseny amb <i>Incorporació tardana</i>	524
3.1	Decisions prèvies	524
	Models diferents segons la creació	524
	Activació de la <i>i:Inscripció</i>	524
	Fixem la diversitat	525
3.2	Anàlisi dels enllaços dirigits d'atribut	525
	Un apartat prescindible	525
3.2.1	Creació de la <i>i:Inscripció</i>	525
	Mecanisme de creació de la <i>i:Inscripció</i>	525
	Enllaç dirigit d'emmagatzematge d' <i>i:Inscripció</i>	525
	Afectació ja coneguda	525
3.2.2	Creació del <i>p:Partícep</i>	526
	Mecanisme de creació del <i>p:Partícep</i>	526
	Enllaç dirigit d'emmagatzematge de <i>p:Partícep</i>	526
	Cap novetat	526
3.2.3	Activació de la <i>i:Inscripció</i>	526
	Activació de la <i>i:Inscripció</i>	526
	Una altra afectació ja coneguda	526
3.3	Canvis en el disseny	526
	Creació de la <i>i:Inscripció</i>	526
	<i>novaInscripcio(c:Caminada)</i>	526
	<i>fiInscripcio()</i>	527
	Localitat d'una visibilitat	527
	Localitat de la <i>i:Inscripció no finalitzada</i>	527
	Activació de la <i>i:Inscripció</i>	527
	Dos termes sinònims	527
	Validesa a la separació dels repositoris	527
	Repositori no accessible	527
	Canvis mínims	527
	Canvis a MComp	528
3.4	Afectació en els contractes contractes	528
	Estabilitat de l'especificació	528
	Completesa expositiva	528
3.4.1	Modificació de l'especificació	528

	Observació dels canvis	529
3.4.2	Anotació del contracte	529
	Contractes sense canvis	529
	Anotació	529

1 Els contractes fins al moment

(127)



2 Anàlisi de la *Incorporació tardana*

Noves realitzacions. Els contractes preveuen les següents noves realitzacions: la i :Inscripció i els seus p :Partíceps.

Enllaços dirigits generats pel disseny. El disseny generarà, com a mínim, els enllaços d'emmagatzematge de i :Inscripció i p :Partícep, i els d'activació.

Anàlisi de creadors. Sintetitzem l'anàlisi de la creació. Els resultats pel cas de la creació del Partícep s'obtenen de la mateixa manera que s'han obtingut els de la Inscripció.[→]

11. *GLS* L'ES novaInscripció, pàgina 199

- Inscripció.
 - Creador canònic: Caminada
 - Creador orb: El controlador K
- Partícep.
 - Creador canònic: Inscripció
 - Creadors orbs: El controlador K , Caminada

Enllaços dirigits que cal analitzar. Afegim ara els resultats obtinguts en l'anàlisi de l'activació.[→]

13. *GLS* POST: La i :Inscripció es manté activa, pàgina 231

- Creació Inscripció: $X \rightarrow \text{Inscripció}^*, X \in \{K, \text{Caminada}\}$
- Activació Inscripció:
 - *Activació directa*: $K \rightarrow \text{Inscripció}$
 - *Activació indirecta*: $K \rightarrow \text{Caminada} \rightarrow \text{Inscripció}$
- Creació Partícep: $Y \rightarrow \text{Partícep}^*, Y \in \{K, \text{Caminada}, \text{Inscripció}\}$

Enllaços dirigits exigits per l'especificació. L'especificació demana lligar la nova i :Inscripció a la c :Caminada donada, i cada nou p :Partícep a la nova i :Inscripció. Però no diu en quin sentit han d'anar els enllaços dirigits. Per tant:

- $c:\text{Caminada} \rightarrow i:\text{Inscripció}^*$ o $i:\text{Inscripció} \rightarrow c:\text{Caminada}$
- $i:\text{Inscripció} \rightarrow p:\text{Partícep}^*$ o $p:\text{Partícep} \rightarrow i:\text{Inscripció}$

Quins d'aquests enllaços queden afectats per la *Incorporació tardana*?

(110) ?

Una resposta clara. El principi de la *Incorporació tardana* en aquest sentit és molt clar: queden afectats els enllaços dirigits $u \rightarrow v$ on u és un objecte preexistent i v és un objecte nou.

Aplicació al nostre cas. Repetim tot seguit els enllaços dirigits que hem vist que cal analitzar, però marcant en blau els objectes nous:

- Creació Inscripció: $X \rightarrow \text{Inscripció}^*$, $X \in \{K, \text{Caminada}\}$
- Activació Inscripció:
 - Activació directa: $K \rightarrow \text{Inscripció}$
 - Activació indirecta: $K \rightarrow \text{Caminada} \rightarrow \text{Inscripció}$
- Creació Partícep: $Y \rightarrow \text{Partícep}^*$, $Y \in \{K, \text{Caminada}, \text{Inscripció}\}$
- Exigències de l'especificació
 - $c:\text{Caminada} \rightarrow i:\text{Inscripció}^*$ o $i:\text{Inscripció} \rightarrow c:\text{Caminada}$
 - $i:\text{Inscripció} \rightarrow p:\text{Partícep}^*$ o $p:\text{Partícep} \rightarrow i:\text{Inscripció}$

(128)

Enllaços dirigits afectats per la Incorporació tardana.

- Visibilitat d'emmagatzematge de la Inscripció.
 - Independentment del creador considerat hi ha afectació
- Visibilitat d'emmagatzematge del Partícep.
 - Només hi ha afectació en el cas d'emprar el creador orb
- Activació Inscripció
 - Independentment de si usem visibilitat d'activació directa o indirecta hi ha afectació
- Exigències de l'especificació
 - El lligam entre *Caminada* i *Inscripció* pot quedar afectat o no, en funció de quin enllaç dirigit emprí el disseny per realitzar aquest lligam
 - El lligam entre *Inscripció* i *Partícep* no queda afectat

3 Un disseny amb Incorporació tardana

3.1 Decisions prèvies

Models diferents segons la creació. Per a la Inscripció tenim un creador orb i un creador canònic; per al Partícep tenim dos creadors orbs i un creador canònic. En total tenim 6 models possibles diferents.

27.3 Un disseny amb *Incorporació tardana* J.M. Merenciano

Activació de la $i : \text{Inscripció}$. Si usem el creador orb d' $i : \text{Inscripció}$ l'activació per visibilitat indirecta no té sentit. En els altres models, però, tant podem mantenir l'activació per visibilitat directa com indirecta. En total tenim 9 models diferents.¹

Fixem la diversitat. L'anàlisi d'aquests 9 models és molt similar. Per tant en el que segueix ens restringirem a un únic model.

El model que considerem es pot descriure com:

- Caminada com a creador canònic d' Inscripció
- Inscripció com a creador canònic de Partícep
- Activació d' Inscripció per la visibilitat indirecta
 $K \rightarrow \text{Caminada} \rightarrow \text{Inscripció}$

(129)

3.2 Anàlisi dels enllaços dirigits d'atribut

Un apartat prescindible. Acabem d'analitzar com pot afectar la *Incorporació tardana* els diferents models possibles. Aquí refem l'anàlisi en el cas concret del model expressat en la conclusió \rightarrow . L'apartat no aporta res de nou; simplement repeteix l'anàlisi a un cas concret i no pas al cas general, en atenció a aquells lectors a qui els casos generals els perden una mica.

Conclusió 129, pàgina 525

3.2.1 Creació de la $i : \text{Inscripció}$

Mecanisme de creació de la $i : \text{Inscripció}$. La $c : \text{Caminada}$ crea i emmagatzema la $i : \text{Inscripció}$. En aquest cas el principi de la *Recuperació genètica* \rightarrow no exigeix cap visibilitat de recuperació o cap visibilitat auxiliar.

???, pàgina ??

Enllaç dirigit d'emmagatzematge d' $i : \text{Inscripció}$. L'enllaç dirigit d'atribut generat com a conseqüència de la creació i emmagatzematge de la $i : \text{Inscripció}$ és realització de la següent visibilitat:

- *Emmagatzematge*. $\text{Caminada} \rightarrow \text{Inscripció}^*$

Afectació ja coneguda. La visibilitat d'emmagatzematge està afectada per la *Incorporació tardana*, cosa que ja sabíem. \rightarrow I es correspon a l'exigència de l'especificació *referent(c,i)*.

Conclusió 128, pàgina 524

¹De model amb creador orb d' Inscripció n'hi ha tres, en funció del creador de Partícep ; i cadascun d'aquests té una sola manera de resoldre l'activació. Per tant, tenim 3 models amb creador orb. Anàlogament de model amb creador canònic d' Inscripció n'hi ha tres, en funció del creador de Partícep ; i cadascun d'aquests tres models té dues maneres de resoldre l'activació. En total 6 models possibles, que sumats als 3 models amb creador orb dona un total de 9 models diferents possibles.

3.2.2 Creació del $p:Partícep$

Mecanisme de creació del $p:Partícep$. Tot seguit fem la descripció d'un model per a la creació del $p:Partícep$. Deixem com a exercici analitzar la bondat d'aquest model. La $i:Inscripció$ crea i emmagatzema el $p:Partícep$. En aquest cas el principi de la *Recuperació genètica*[→] no exigeix cap visibilitat de recuperació o cap visibilitat auxiliar.

Enllaç dirigit d'emmagatzematge de $p:Partícep$. L'enllaç dirigit d'atribut generat com a conseqüència de la creació i emmagatzematge del $p:Partícep$ és realització de la següent visibilitat:

- *Emmagatzematge*. $Inscripció \rightarrow Partícep^*$

Cap novetat. La visibilitat d'emmagatzematge *no* està afectada per la *Incorporació tardana*, ja que els enllaços dirigits generats són sortints d'un objecte nou, la $i:Inscripció$, i tal com ja havíem vist anteriorment. [→] Per l'altra banda cal observar com aquesta visibilitat d'emmagatzematge es correspon a l'exigència de l'especificació *inscrit*(i,p).

3.2.3 Activació de la $i:Inscripció$

Activació de la $i:Inscripció$. El contracte exigeix deixar activa la $i:Inscripció$ així que es crea, i mantenir-la activa fins a rebre *fiInscripcio*(\cdot). I aquesta activació la fem amb la visibilitat indirecta $K \rightarrow Caminada \rightarrow Inscripció$.

Una altra afectació ja coneguda. Evidentment, i tal com ja sabíem, la visibilitat $Caminada \rightarrow Inscripció$ queda afectada per la *Incorporació tardana*.

(130)

Enllaços afectats per la *Incorporació tardana*.

- $Caminada \rightarrow Inscripció^*$
- $Caminada \rightarrow Inscripció$

3.3 Canvis en el disseny

Creació de la $i:Inscripció$. En el disseny original la $c:Caminada$ crea i emmagatzema la $i:Inscripció$. Ara, amb la *Incorporació tardana*, la $c:Caminada$ també crea i emmagatzema la $i:Inscripció$, però ambdues responsabilitats les resol en moments diferents.

***novaInscripcio*($c:Caminada$).** En aquest ES la $c:Caminada$ crea la $i:Inscripció$, i enlloc d'emmagatzemar-la al repositori $Caminada \rightarrow Inscripció^*$ l'emmagatzema en un repositori temporal al CU. Aquest repositori conté totes les $i:Inscripcions$ incompletes durant el CU; però com que d'aquestes

27.3 Un disseny amb *Incorporació tardana* J.M. Merenciano

només en tenim una això significa que l'emmagatzematge és amb la visibilitat $\text{Caminada} \rightarrow \text{Inscripció}$.

fiInscripcio(). Ara és el moment d'afegir la $i : \text{Inscripció}$, ja acabada, al repositori $\text{Caminada} \rightarrow \text{Inscripció}^*$, i treure'la del repositori $\text{Caminada} \rightarrow \text{Inscripció d'} : \text{Inscripcions incomplertes}$.

Localitat d'una visibilitat. Com veiem, l'enllaç dirigit $c \rightarrow i$ és local al CU: es genera en el primer ES, i s'elimina en el darrer ES. És evident, per tant, que la visibilitat $\text{Caminada} \rightarrow \text{Inscripció}$ és el repositori de les $i : \text{Inscripcions}$ no finalitzades.

Localitat de la $i : \text{Inscripció}$ no finalitzada. La $i : \text{Inscripció}$ acabada de crear és una $i : \text{Inscripció no finalitzada}$. I aquest estat és local al CU de creació que estem considerant: la $i : \text{Inscripció}$ activa és *no finalitzada* mentre no es rebí l'ES *fiInscripcio()*.

Activació de la $i : \text{Inscripció}$. Hem decidit que aquesta activació es manté amb la visibilitat indirecta $K \rightarrow \text{Caminada} \rightarrow \text{Inscripció}$. I l'especificació diu que l'activació la nova $i : \text{Inscripció}$ s'activa en el moment de crear-la, i es desactiva en el darrer ES.

Dos termes sinònims. La $i : \text{Inscripció}$ activa és aquella que estem construint, i per tant és una $i : \text{Inscripció no finalitzada}$. I no podem tenir inscripcions *no finalitzades* que no siguin la que estem construint, i que per tant està activa. En conclusió, en aquest CU, *activa* és sinònim de *no finalitzada*.

En aquest CU dir que la $i : \text{Inscripció}$ està *activa*, o dir que està en l'estat *no finalitzada* és dir exactament el mateix

(131)

Validesa a la separació dels repositoris. L'aplicació de la *Incorporació tardana* exigeix tenir dos repositoris: el local al CU i el permanent. Però, en aquest cas, aquesta duplictat de repositori no és cap exigència forassenyada: quan en l'especificació se'ns demana l'activació de la $i : \text{Inscripció}$ ja estem dient que ens cal, a nivell de disseny, una visibilitat monoavaluada. En d'altres paraules: en aquest cas l'aplicació de la *Incorporació tardana* només canvia el disseny en una petita qüestió: el moment en què es fa l'add de la $i : \text{Inscripció}$ al repositori $\text{Caminada} \rightarrow \text{Inscripció}^*$.

Repositori no accessible. Una observació final: el repositori $d' : \text{Inscripcions no finalitzades}$, $\text{Caminada} \rightarrow \text{Inscripció}$ només té sentit si dins del CU és accessible. Això és, que dins del CU el controlador tingui visibilitat de la $c : \text{Caminada}$. I això és el que assegura la visibilitat indirecta d'activació $K \rightarrow \text{Caminada} \rightarrow \text{Inscripció}$.

Canvis mínims. Els canvis en els contractes, en aquest cas, afecten ben poc el disseny. Per una banda cal emmagatzemar temporalment, i de manera local al CU, la *i : Inscripció* creada. Per altra banda cal afegir la *i : Inscripció* als repositoris del sistema quan es rebi l'ES *fiInscripcio()*.

Canvis a MComp. El MComp no ha variat: les responsabilitats, les visibilitats i els acoblaments són els mateixos; el que ha canviat és el moment en què es demana resoldre determinada responsabilitat (la incorporació de la nova *i : Inscripció* als repositoris del sistema).

3.4 Afectació en els contractes

Estabilitat de l'especificació. El principi de l'*Estabilitat de l'especificació*[→] diu que no hem de tocar els contractes. La *Incorporació tardana* només té sentit en el disseny perquè fa referència a l'emmagatzematge. Per tant n'hi ha prou en fer una anotació.

Completesa expositiva. Malgrat això, en el següent apartat plantejarem com quedarien els contractes si els volguéssim canviar. Però recordem, no s'han de canviar.

3.4.1 Modificació de l'especificació

Plantegem com quedaria la modificació de l'especificació. Però el principi de l'*Estabilitat de l'especificació*[→] diu que **NO** hem pas de modificar l'especificació per mor de la *Incorporació tardana*

! (111)

Contracte <i>novaInscripcio(c : Caminada)</i>
<p>Paràmetres</p> <ul style="list-style-type: none">• <i>c</i>: Caminada a la que es vol fer la inscripció <p>PRE</p> <ol style="list-style-type: none">1. No hi ha cap <i>Inscripció</i> activa <p>POST</p> <ol style="list-style-type: none">1. Hi ha una nova realització <i>i : Inscripció</i>2. \emptyset3. La inscripció <i>i</i> està activa
Contracte <i>fiInscripcio()</i>
<p>PRE</p> <ol style="list-style-type: none">1. Hi ha una <i>i : Inscripció</i> activa <p>POST</p> <ol style="list-style-type: none">1. Hi ha un enllaç entre <i>i</i> i <i>c</i>, corresponent a l'associació <i>relativa a</i>

27.3 Un disseny amb *Incorporació tardana*
J.M. Merenciano

2. No hi ha cap *Inscripció activa*

Observació dels canvis. Com es pot veure, en aquest cas, la reconsideració de l'especificació com a conseqüència de l'aplicació de la *Incorporació tardana*, només fa canviar una PRE d'ES.

3.4.2 Anotació del contracte

Això és el que cal fer, no pas el que hem fet en el darrer apartat

(112) !

Contractes sense canvis. Els contractes són els mateixos que teníem abans d'aplicar la *Incorporació tardana*. No hi ha hagut canvis.

Anotació. Ara n'hi ha prou en introduir un anotació en els contractes. Tot seguit mostrem les anotacions que tenim fins wel moment, i amb blau destaquem la nova anotació:

Anotacions als contractes.

- La *:Inscripció* es manté activa amb la visibilitat indirecta *K→Caminada→Inscripció*
- L'emmagatzematge de la *:Inscripció* (la visibilitat *Camina-da→Inscripció**) està afectat per la *Incorporació tardana*

(132)

Capítol 28

GLS Cas d'ús *Creació de dorsals*

1	Esdeveniment <i>activaCaminada (nomC)</i>	533
1.1	Contracte de <i>activaCaminada (nomC)</i>	533
1.2	Disseny de <i>activaCaminada (nomC)</i>	533
2	Esdeveniment <i>creaDorsalsGrup (id)</i>	535
2.1	Contracte de <i>creaDorsalsGrup (id)</i>	535
2.2	Controlador	535
2.3	Anàlisi de l'argument	535
2.4	Creador	538

Contingut detallat del capítol 28

1	Esdeveniment <i>activaCaminada (nomC)</i>	533
1.1	Contracte de <i>activaCaminada (nomC)</i>	533
1.2	Disseny de <i>activaCaminada (nomC)</i>	533
	Responsabilitats per resoldre	533
	Necessitats de l'encaixador	533
	Robustesa de la identificació	533
	Candidat a <i>encaixador</i>	533
	Com mantenir la <i>c:Caminada activa</i>	533
	Aplicació de <i>Baix acoblament</i>	533
	Assignació d'ambdues responsabilitats	533
	Independència del controlador	534
	Necessitats comunicatives entre els ES	534
	Cohesió en la compartició del controlador	534
	De moment, simplicitat	534
	Excursió. (Compartició de controladors)	534
	Amatents a la possible reconsideració	535
2	Esdeveniment <i>creaDorsalsGrup (id)</i>	535
2.1	Contracte de <i>creaDorsalsGrup (id)</i>	535
2.2	Controlador	535
2.3	Anàlisi de l'argument	535
2.3.1	Resolució de l'encaix	535
	Necessitats de l'encaixador	535
	Àmbit d'identificació	536
	Excursió. (No coincidència d'àmbits.)	536
	El controlador com a encaixador	536
	Validesa de l'encaixador genèric	536
	Delegació de l'encaix	536
	Encaixador executor	536
	Dues alternatives d'anàlisi	537
2.3.2	Àmbits i usabilitat	537
	Visibilitat de l'àmbit d'identificació	537
	Comprovació d'identificador	537
2.4	Creador	538
2.4.1	Emmagatzemadors canònics	538
	Emmagatzemador canònic	538
	Encaixador executor	538
2.4.2	Experts en la creació del <i>Dorsal</i>	538
2.4.3	El concepte <i>Dorsal</i>	538
	Requeriments	538
	Àmbits d'unicitat	538
	Identificació dels dorsals	538
	Visibilitat de <i>GenIdDorsal</i>	539
	La <i>Caminada</i> com a executora	539

1 Esdeveniment *activaCaminada (nomC)*

1.1 Contracte de *activaCaminada (nomC)*

Contracte <i>activaCaminada (nomC)</i>	
Paràmetres	
• <i>nomC</i> :	Nom caminada
PRE	
1.	Hi ha una <i>c:Caminada</i> amb <i>c.nom=nomC</i>
POST	
1.	<i>c:Caminada</i> esdevé activa

1.2 Disseny de *activaCaminada (nomC)*

Responsabilitats per resoldre. L'ES demana l'activació d'una *c:Caminada*; la *c:Caminada* que cal activar s'indica mitjançant el seu identificador. Tenim doncs dues responsabilitats: resoldre l'encaixada i deixar com a activa la realització recuperada.

Necessitats de l'encaixador. El principi de l'*Encaixada* demana resoldre en primer lloc la *recuperació* de l'objecte. Per fer-ho cal que l'encaixador tingui visibilitat de totes les *Caminades* del sistema.

Robustesa de la identificació. El principi de la *Identificació robusta* exigeix un mecanisme de *comprovació d'identificador*; el principi dels *Àmbits accessibles* demana, per tal d'oferir aquest mecanisme, la visibilitat des del controlador de l'àmbit d'identificació pertinent (les *Caminades* del sistema en el nostre cas).


Candidat a encaixador. Com hem vist,¹ un bon candidat a resoldre l'encaix és el controlador, malgrat que llavors violem el principi de l'*Encarrilament*. L'aventatge és que en usar el controlador com a encaixador les visibilitats necessàries estan assegurades pel principi dels *Àmbits accessibles*.

Com mantenir la *c:Caminada* activa. Un cop resolt l'encaix cal deixar activa la *c:Caminada* recuperada; i això significa donar-li visibilitat (indirecta) monoavaluada des del controlador.

Aplicació de *Baix acoblament*. El menor acoblament s'obté assignant les responsabilitats de la recuperació i la de mantenir activa la *c:Caminada* recuperada a un mateix component. Altrament el responsable de la recuperació (l'encaixador) ha de *comunicar* la *c:Caminada* al responsable de mantenir-la activa (algú que sigui destinació d'un enllaç dirigit indirecte monoavaluat que té l'origen en el controlador).


¹Vegeu apartat 17.4.2.Resoldre l'encaixada, pàgina 353.

Assignació d'ambdues responsabilitats. De tot plegat, si admetem la violació del principi de l'*Encarrilament*, el *controlador* rep la responsabilitat de recuperar la *c*:Caminada i de mantenir-la activa mitjançant un atribut.

 (113)


El controlador resol l'encaixada i manté la *c*:Caminada activa

Independència del controlador. És important d'observar com la decisió acabada de pendre és del tot independent al controlador considerat.

 (114)

Quin controlador usem?


Necessitats comunicatives entre els ES. L'ES *activaCaminada (nomC)* té per funció deixar les coses preparades pel següent ES d'aquest CU; en concret per *creaDorsalsGrup (id)*. Per tal de poder comunicar ambdós ES caldrà usar el principi *Controlador de CU*.

 (115)

Podem usar el mateix objecte controlador *g:GLS* com a controlador, tant del CU *ferInscripció* com del CU *crearDorsals*?

Cohesió en la compartició del controlador. En fer-ho el controlador disminueix la seva *cohesió*. Malgrat tot, si admetem que la creació de *dorsals* és un procés intrínsecament lligat a la introducció d'*inscripcions*,² llavors podem considerar que la *cohesió* es manté en uns nivells admissibles.

De moment, simplicitat. Mentre la cohesió no degeneri excessivament ens decantem pel model més simple. I per tant escollim compartir el controlador.

 (116)

Prenem *g:GLS* com a controlador dels CU *ferInscripció* i *crearDorsals*

Excursió. (Compartició de controladors) Ambdós CU, *ferInscripció* i *crearDorsals*, necessiten un mecanisme de comprovació d'identificador, que hem resolt amb l'ES *existeixCaminada (nom)*. Si els controladors pertinents són realitzacions d'un mateix component (*GLS* en el nostre

²Aquesta afirmació és del tot discutible. De fet pot ser que un mateix canviï de parer respecte ella al llarg del desenvolupament.

cas) aconseguim no haver de tenir dos mètodes diferents amb la mateixa funcionalitat (a nivell d'implementació, tenim compartició de codi).

Tant en un CU com en l'altre necessitem tenir accés al repositori de totes les *Caminades*. Per assegurar la consistència de la visibilitat d'ambdós controladors, és a dir, que vegin exactament les mateixes *Caminades*, el més simple és assegurar que són el mateix controlador. (A nivell d'implementació, tenim compartició de dades).

Un mecanisme per assegurar la compartició del controlador (sense complicar el disseny) i alhora desacoblar el disseny d'ambdós CU, és convertir GLS en un *Singleton*. Observem a més que aquesta decisió pot venir ja de l'especificació: el nom *GLS* (i no *Empresa* o *Delegació*, per exemple) ja denota una única realització del concepte.

Amatents a la possible reconsideració. La decisió presa sobre el controlador ha de ser mantinguda amb precaució. En virtut del principi de la *Reconsideració contínua* caldrà estar sempre amatents a un possible canvi de criteri. Per exemple, si el disseny del CU *crearDorsals* acaba sent complex, pot ser convenient desacoblar les responsabilitats dels dos CU; la qual cosa passaria per assignar-los controladors diferents.

2 Esdeveniment *creaDorsalsGrup(id)*

2.1 Contracte de *creaDorsalsGrup(id)*

Contracte <i>creaDorsalsGrup(id)</i>	
Paràmetres	
• <i>id</i> :	Identificador del grup, és a dir, de la inscripció
PRE	
1.	Hi ha una <i>c:Caminada</i> activa
2.	Hi ha una <i>i:Inscripció</i> corresponent a la <i>c:Caminada</i> amb <i>i.id=id</i>
POST	
1.	S'han creat els dorsals de tots els participants del grup <i>i:Inscripció</i>

2.2 Controlador

Les necessitats comunicatives entre la POST de *activaCaminada(nomC)* i la PRE de *creaDorsalsGrup(id)* ens ha portat a usar un mateix controlador per tot el CU.

2.3 Anàlisi de l'argument

2.3.1 Resolució de l'encaix

Necessitats de l'encaixador. El primer que cal és resoldre l'encaixada de la *Inscripció*. El contracte de *creaDorsalsGrup(id)* ens diu que l'argument rebut és l'identificador de la *i:Inscripció* dins la *c:Caminada* activa;

Àmbit de resolució de l'encaixada

és a dir, l'àmbit de resolució de l'encaixada és *c*. Per tant l'encaixador ha de tenir visibilitat de les Inscripcions associades a la *c*: Caminada activa.

Àmbit d'identificació. L'àmbit d'identificació de les Inscripcions és la Caminada.³ Per tant l'àmbit de resolució de l'encaixada coincideix amb l'àmbit d'identificació.

Excursió. (No coincidència d'àmbits.) Suposem que l'àmbit d'identificació fos tot el sistema. És a dir, que existeix un repositori centralitzat d'Inscripcions (i per tant, segurament el propi controlador seria el creador d'Inscripcions). Llavors, per tal de poder recuperar la Caminada, cada Inscripció hauria de recordar a quina Caminada es correspon. A més, per fer la recuperació de la Inscripció caldria fer una cerca en el repositori centralitzat d'Inscripcions usant l'identificador de la donat com a argument de l'ES, i tot seguit comprovar que la *i*: Identificació recuperada es correspon a la Caminada activa. Tot plegat sembla un pèl massa rebuscat.

Una alternativa possible seria considerar que el repositori centralitzat d'Inscripcions té una rèplica fragmentada per Caminades. Llavors totes les decisions preses tenint en compte l'àmbit d'identificació (com la possibilitat que el controlador sigui el creador d'Inscripcions) continuen essent vàlides; però a més ara tenim la possibilitat de delegar en la *c*: Caminada activa la responsabilitat de resoldre l'encaixada. La situació, doncs, és similar al cas en el que l'àmbit de recuperació i l'àmbit d'identificació coincideixen: l'encaixador és qui té visibilitat estricta⁴ de l'àmbit de l'encaix.

El controlador com a encaixador. De moment el controlador ja té la responsabilitat de resoldre l'encaixada de Caminades; si li assignem també la responsabilitat de resoldre l'encaixada de la Inscripció no disminuïm la cohesió: sempre podem considerar que hem assignat al controlador el paper d'encaixador genèric.

Encaixador genèric

Validesa de l'encaixador genèric. Com hem vist en estudiar les diferents maneres de resoldre les encaixades,⁵ l'ús del controlador com a encaixador viola el principi de l'Encarrilament, però tot i així ho podem considerar com una alternativa vàlida davant dels problemes de les altres possibilitats.⁶

Delegació de l'encaix. Ara bé, de fet la visibilitat que el controlador té de l'àmbit de l'encaixada és una visibilitat indirecta a través de la *c*: Caminada activa. És a dir, que el controlador per resoldre l'encaixada de la Inscripció que exigeix l'ES *creaDorsalsGrup(id)* ha de delegar la tasca de l'encaixada a la *c*: Caminada.

³Decisió 83, pàgina 424.


⁴Diem que *a:A* té visibilitat estricta del conjunt *B* si el conjunt de realitzacions $\{b:B \mid b \text{ és visible des de } a:A\}$ és exactament el conjunt *B*.

⁵Vegeu l'apartat 17.4.2. Resoldre l'encaixada, pàgina 353.

⁶El principi de la Solidesa argumentada, 355, justament ens diu que l'important no és complir al peu de la lletra els principis, sinó el tenir clar què és el que fem en cada moment.

Encaixador executor. Un cas particular de delegació de l'encaix és l'encaixador executor. En aquest cas la *c:Caminada* no només resoldria l'encaix, sinó que també s'encarregaria d'enviar a la *i:Inscripció* recuperada els missatges necessaris per a crear els dorsals.

La *c:Caminada* activa és la responsable de resoldre l'encaix

(117) 

Dues alternatives d'anàlisi. Tot i que en l'anàlisi genèrica sobre l'encaix⁷ donàvem prioritat a la delegació de l'encaix per sobre de l'encaixador executor, mantindrem de moment obertes ambdues possibilitats. Com sempre, l'objectiu és la completesa expositiva i la possibilitat de veure en un cas concret allò que prèviament hem estudiat en abstracte.

A *c:Caminada* li deleguem l'encaix; però li deleguem també l'execució de les responsabilitats de l'ES?


(118) ?

2.3.2 Àmbits i usabilitat

Visibilitat de l'àmbit d'identificació. El principi dels *Àmbits visibles* exigeix que el controlador tingui visibilitat, directa o indirecta, de l'àmbit d'identificació. La PRE que demana una *Caminada* activa, i la fragmentació del repositori d'*Inscripcions* en *Caminades*, asseguren les visibilitat exigides.

Comprovació d'identificador. El principi de la *Identificació usable* exigeix l'existència d'un mecanisme de comprovació d'identificador. Si admetem que com a identificador qualsevol seqüència de lletres⁸ és correcta, n'hi ha prou amb un mecanisme de *comprovació d'existència*.

Intruduíu l'ES *existeixInscripció(id)*

(119) 

Contracte <i>existeixInscripció(id)</i>	
Paràmetres	
• <i>id:</i>	Identificador de la inscripció
PRE	

⁷Vegeu l'apartat 17.4.2. Resoldre l'encaixada, pàgina 353.

⁸L'enunciat diu que els dorsals contenen una seqüència de lletres que coincideix amb el resguard de la inscripció. El que fem aquí és assumir que l'identificador del resguard i l'identificador de la *i:Inscripció* és el mateix.

1. Existeix una *c:Caminada* activa

POST

 1. Cert \Leftrightarrow En el sistema existeix una *i:Inscripció* “corresponent” a *c* amb *i.nom=nom*
 2. La *c:Caminada* es manté activa

2.4 Creador

@@@@@@

2.4.1 Emmagatzemadors canònics

Emmagatzemador canònic. L'únic emmagatzemador canònic és la *Inscripció*. Per tant n'hi ha prou en assegurar que també és *experta* en la creació per poder-li donar el paper de *creador canònic*.

Encaixador executor. En cas que realment la *Inscripció* sigui el *creador canònic* dels *Dorsals*, llavors pren força l'encaixador executor: la *Caminada* resol

—¿ si la caminada te una inscripció activa li pot delegar la tasca

2.4.2 Experts en la creació del *Dorsal*

—¿ Cal estudiar què carai és la info del dorsal, i quin és l'identificador

2.4.3 El concepte *Dorsal*

Requeriments. Els requeriments diuen que el *Dorsal* té un número únic, i una seqüència de lletres que identifica el grup i coincideix amb l'identificador de la *i:Inscripció*.


? (120)

Quin és l'àmbit d'unicitat del número dels dorsals?

Àmbits d'unicitat. L'àmbit d'unicitat pot ser la *c:Caminada* o tot el sistema. Sigui com sigui podem usar el número dels dorsals com a identificadors d'aquests dins l'àmbit d'unicitat pertinent. En el cas d'usar unicitat global direm que tenim *dorsals amb identificació global*; en el cas d'usar la *Caminada* com a àmbit d'unicitat, direm que tenim *dorsals amb identificació local*.

Identificació dels dorsals. Els *Dorsals* són d'*identificació interna*. De cara a la cohesió, mantenibilitat i canvis en els sistemes de generació, decidim, seguint els principis de la *Fabricació pura* i la *Independència algorísmica*, introduir un component específic per generar el valor d'identificació.

La responsabilitat de generar el número únic del dorsal l'assignem per a un component específic *GenIdDorsal*.

(121) 

Visibilitat de *GenIdDorsal*. El component *GenIdDorsal* ha de ser visible pel creador del *Dorsal*

El resultat seria una minva important de cohesió de la *Caminada*, que passaria a tenir les següents responsabilitats:

- Creadora i emmagatzemadora d'*Inscripcions*
- Encaixadora d'*Inscripcions*
- Creadora dels dorsals

La *Caminada* com a executora. Plantejar-se la *Caminada* com a executora de les responsabilitats de *creaDorsalsGrup(id)* no és només un problema de cohesió. Sense entrar en detalls, segurament caldrà crear *Dorsals*, i emmagatzemar-los. És la *Caminada* un *creador canònic*? La resposta és no.⁹

⁹En alguna versió de MC pot ser que hi hagi una associació entre *Caminada* i *Dorsal*, la qual cosa permetria usar la *Caminada* com a creador canònic de *Dorsal*, però sembla evident que el MC més clar i simple és el que té una associació entre *Participació* i *Dorsal*.

Historial de versions

Versió 1.1 (Febrer de 2017)

Propietats de consistència^{Canvi menor}

S'introdueix una formulació general per a les **propietats de consistència**.

Emmagatzemador^{Canvi important}

- Nous principis.
 - S'introdueixen els principis *Emmagatzemador selectiu* i *Emmagatzemador general*
 - * El principi *Emmagatzemador* tal i com estava a la versió 1.0 només tenia en compte la no violació d'*Espill* en el nivell de les abstraccions.
 - * Aquí introduïm el nivell de les realitzacions. Com a conseqüència apareixen dos principis nous, i cal refinar el principi existent
 - Es modifica *Emmagatzemador (canònic)*
 - * Contempla la no violació d'*Espill* a nivell de les realitzacions
 - * És la nova versió del principi *Emmagatzemador* de la versió 1.0
 - S'introdueix en forma de principi *Emmagatzemador generalitzat* la jerarquia d'aplicació dels principis de creació
- **Creador canònic**. Es reformula tenint en compte els nous principis d'emmagatzematge

Versió 1.0 (juny de 2016)

- **Llegenda**. S'intordueix la llegenda amb els diferents elements discursius i les seves marques identificatives.
- **Canvi de format**. Canvio tots els formats donant-ne un aspecte més visual gràcies `tcolorbox`
- Revisió fins el capítol **Àmbis d'identificació**, exclòs

- En aquesta revisió es detecten alguns problemes en la definició d'**identificador** i **repositori**. D'aquí la necessitat de la versió 1.1
- **Referències creuades**. Les traiem del peu de pàgina, i les posem en una nota al marge
- **Icones**. S'introdueixen icones pels diferents elements remarcats. L'objectiu és millorar-ne la seva localització en una lectura ràpida.

Principis

- Àmbits accessibles, 381
- MOMO: Modisme de l'Objecte
 - MultiObjecte, 100
- Alta cohesió (GRASP), 127
- Aquest és un principi, ix
- Baix Acoblament (GRASP), 128
- Cistella, 352
- Comunicació amb identificadors, 341
- Connexions, 345
- Consistència individualitzadora, 377
- Contracte robust, 330
- Controlador (GRASP), 112
- Controlador de CU, 114
- Creació complerta, 400
- Creador (GRASP), 158
- Creador canònic, 175, 178
- Creador delegat, 160
- Creador generalitzat, 182
- Creador orb, 163
- Desconeixement del grup, 404
- Desenvolupament sota contracte, 14
- Eficiència en segon pla, 228
- Emmagatzemador (canònic), 172
- Emmagatzemador general, 171
- Emmagatzemador selectiu, 168
- Encaixada, 352
- Encapsulament, 11
- Encarrilament, 115
- Espill, 8
- Estabilitat d'identificació, 380
- Estabilitat de l'especificació, 227
- Expert (GRASP), 110
- Franquícia, 9
- Franquícia obligada, 9
- Identificació atributiva, 380
- Identificació referencial, 353
- Identificació robusta, 329
- Identificadors i Presentació, 341
- Igualtat d'oportunitats, 25
- Incorporació tardana, 514
- Individualització efectiva, 416
- Interfície obligada, 12
- Modelització continua, 10
- Ocultació, 11
- Ocultació total, 14
- Reconsideració, 27
- Referent únic, 9
- umlaut itat, 25
- Solidesa argumentada, 355
- Teranyina, 28

Definicions

Àmbit d'identificació, 376

Acoblament, 128

Això és una definició, ix

Anòmia, 266

Associació d'identificació, 369

Atribut de compleció, 507

Cadena d'enllaços dirigits, 63

Caixa blanca, 14

Caixa negra, 15

Cohesió, 127

Completesa, 259

Component, 15

Component ben definit, 296

Concepte feble, 368

Concepte fort, 369

Context d'emissió d'un missatge, 42

Context de localitat, 46

Contracte, 12

Contracte PRE/POST, 13

Creació d'ús, 140

Creador canònic, 176, 181

Creador orb, 163

Creador orb pur, 182

Delegació, 68

Desenvolupament per capes (primera aproximació), 337

Desenvolupament per capes (proposta definitiva), 340

Emmagatzemador canònic, 172

Emmagatzemador general, 171

Emmagatzemador selectiu, 169

Emmagatzematge primari, 176

Enllaç (dirigit) d'emmagatzematge, 146

Enllaç dirigit, 40

Enllaç dirigit (generalitzat), 59

Enllaç dirigit d'atribut, 45

Enllaç dirigit de paràmetre, 48

Enllaç dirigit local, 46

Enllaç dirigit monoavaluat o simple, 57

Enllaç dirigit multiavaluat, 57

Enllaços dirigits indirectes, 62

Enllaços incidents i sortints, 513

Espai d'identificació, 290

Especificació per capes, 339

Estat de compleció, 507

Expert per delegació, 159

Fragment d'un repositori, 149

Identificació externa, 326

Identificació interna, 326

Identificació mixta, 326

Identificador d'una realització, 290

Informació d'expertesa, 159

Interfície, 12

Isomorfia entre model i realitat, 262

Mecanisme de comprovació
d'identificador, 327

Migració de realitzacions, 378

Model, 8

Model coherent, 262

Model d'un conjunt d'entitats, 294

Multiobjecte, 90

Nom, 264

Objecte, 36

Propietat atributiva, 114

Propietat d'identificació, 289

Propietat d'identificació especular, 294

Propietat d'identificació textual, 291

Rèplica d'un repositori, 379

Realització activa, 222

Realització compartida, 148, 378

Recuperació exhaustiva, 323

Recuperació referencial, 324

Repositori, 147

Repositori centralitzat, 148

Repositori de contrast, 396

umlaut it, 148

Repositori replicat, 379

Repositoris disjunts, 148

Resolució de l'encaixada, 353

Restricció d'una propietat
d'identificació, 307

Sinonímia, 269

Sistema de tres capes, 338

Solidesa, 259

Viabilitat indirecta d'atribut, 67

Viabilitat indirecta de paràmetre, 67

Visibilitat (directa) d'un objecte, 52

Visibilitat d'activació, 222

Visibilitat d'atribut, 53, 56

Visibilitat d'emmagatzematge, 146
Visibilitat de paràmetre, 53, 56
Visibilitat entre components, 54
Visibilitat generalitzada (d'objecte), 61
Visibilitat generalitzada (d'un
component), 62
Visibilitat indirecta, 65
Visibilitat indirecta local, 67
Visibilitat indirecta monoavaluada, 68
Visibilitat indirecta multiavaluada, 68
Visibilitat local, 53, 57
Visibilitat multiavaluada (d'un
component), 61
Visibilitat multiavaluada (d'un objecte),
60

